昭和64年1月1日発行(毎月1回1日発行)第8巻1号通巻81号 昭和58年11月2日第三種郵便物認可
ISSN0910-7614

パーソナルコンピュータ・マガジン
MZシリーズ,X1/turbo,X68000&ポケコン

Oh!X

オー!エックス 定価540円

特集
いきなり初春(はる)からハードウェア
デジタル回路の基礎知識
X1turboのメモリ拡張
X68000に64180ボードをつなぐ

X68000のライバルは誰か?
ようこそ,メガドライブ!!

MZ-2500ゲームブック作成ツール
Hyper Game Book

S-OS全機種共通システム
パズルゲームLAST ONE/FLICK

LIVE in'89
X1/turboエンデューロ レーサー
X68000「アルルの女」よりファランドール

猫とコンピュータ/知能機械概論
Z80マシン語ゲーム工房/C調言語講座PRO-68K

THE SOFTOUCH SPECIAL
1988年度GAME OF THE YEAR
ノミネート作品発表

1
JAN.1989

■本体＋キーボード＋マウス・トラックボール
CZ-611C-GY(グレー)・-BK(ブラック)標準価格399,800円
写真はCZ-611C-GY＋CZ-601D-GY＋CZ-6ST1-E

■本体＋キーボード＋マウス・トラックボール
CZ-601C-GY(グレー)・-BK(ブラック)標準価格319,800円
写真はCZ-601C-BK＋CZ-603D-BK

〈X68000ACEシリーズの主な特長〉●実装密度をさらに追求して信頼性を高めたマンハッタンシェイプ●68000搭載●テキスト、グラフィック、スプライト、独立3画面設計、最大12Mバイトの大容量メモリ(標準1Mバイト)●フレンドリーOS、Human 68k搭載●連文節変換、マルチフォントをサポートした強力日本語処理●1024×1024ドット(最大表示エリア768×512ドット)の実画面エリアを装備した高解像度表示能力●512×512ドット、65,536色同時発色●水平32、1画面128のパワフルなスプライト機能●オーバースキャン機能を採用した512×512ドットレベルのスーパーインポーズ●テキストビットマップ方式採用●8重和音ステレオFM音源搭載●音声デジタイズ記録AD PCM※採用●マウス・トラックボール標準装備●5インチ1MバイトFDD2基搭載●「X-BASIC」、「辞書ディスク」と各種ユーティリティ、「日本語ワードプロセッサ」をバンドル●20Mバイトハードディスク内蔵(CZ-611C)

※Adaptive Differential PCM

■15型カラーディスプレイテレビ(ドットピッチ0.39mm)CZ-601D-GY(グレー)・-BK(ブラック)標準価格119,800円
■15型カラーディスプレイテレビ(ドットピッチ0.31mm)CZ-611D-GY(グレー)・-BK(ブラック)標準価格145,000円
■14型カラーディスプレイ(ドットピッチ0.31mm) CZ-603D-GY(グレー)・-BK(ブラック) 標準価格84,800円(チルトスタンド同梱)
■チルトスタンドCZ-6ST1-E(グレー)・-B(ブラック)標準価格5,800円(CZ-601D/611D用)

アートの領域へ。

クォリティを維持しつづけることは、ある意味では創造することより困難なこととも言われています。出会いが印象的であればあるほど、その後が大変です。このことは、そのままX68000の歩みを言い得ているかも知れません。確かに技術は日進月歩です。しかしそれだけでコンピュータがもつべき創造性を論ずることはできないのも、また事実です。私たちはテクノロジーとクリエイティブマインド、いわば人とマシンとのソフトウェアインターフェイスで応えます。ホリゾンタルなマシンとしての熟成。そこからはいくつもの分野が見えてくるはずです。そしてどんな分野にしろX68000の仕事はアートであるべきです――。ますます洗練されて信頼性を高めたACEシリーズの登場で、あなたはまた新たな可能性に出会えそうです。

豊富な周辺機器がクリエイティブワークをサポート。

品名	型番	価格
●21型カラーディスプレイ	CU-21CD	標準価格139,800円
●RGBシステムチューナー	CZ-6TU	標準価格 35,800円
●15型カラーディスプレイ	CU-15M1-E	標準価格 99,800円
●カラーイメージスキャナ※1	CZ-8NS1	標準価格188,000円
●カラーイメージユニット※2	CZ-6VT1	標準価格 69,800円
●カラービデオプリンタ	CZ-6PV1	標準価格198,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK7	標準価格122,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK8	標準価格152,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK9	標準価格 89,800円
●熱転写カラー漢字プリンタ	CZ-8PC3	標準価格 65,800円
●熱転写カラー漢字プリンタ	CZ-8PC2	標準価格 69,800円
●ハードディスクユニット(20MB)	CZ-620H	標準価格178,000円
●モデムユニット※3	CZ-8TM2	標準価格 49,800円
●RS-232Cケーブル(平行接続型)	CZ-8LM1	標準価格 7,200円
●RS-232Cケーブル(クロス接続型)	CZ-8LM2	標準価格 7,200円
●拡張I/Oボックス(4スロット)	CZ-6EB1	標準価格 88,000円
●1MB増設RAMボード(内蔵用)	CZ-6BE1A	標準価格 38,000円
●2MB増設RAMボード※4	CZ-6BE2	標準価格 79,800円
●4MB増設RAMボード※4	CZ-6BE4	標準価格138,000円
●FAXボード	CZ-6BC1	標準価格 79,800円
●MIDIボード	CZ-6BM1	標準価格 26,800円
●GP-IBボード	CZ-6BG1	標準価格 59,800円
●ユニバーサルI/Oボード	CZ-6BU1	標準価格 39,800円
●増設用RS-232Cボード(2チャンネル)	CZ-6BF1	標準価格 49,800円
●数値演算プロセッサボード	CZ-6BP1	標準価格 79,800円
●スキャナ用パラレルボード	CZ-6BN1	標準価格 29,800円
●システムラック	CZ-6SD1	標準価格 44,800円
●アンプ内蔵スピーカーシステム(2本1組)	AN-160SP	標準価格 59,800円
●マウス	CZ-8NM2A	標準価格 6,800円
●トラックボール	CZ-8NT1	標準価格 13,800円
●ジョイカード	CZ-8NJ1	標準価格 1,700円
●高性能CRTフィルター	BF-68PRO	標準価格 19,800円

※1 使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ転送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1で接続してください。※2 使用に際してはコンピュータ本体と専用15型カラーディスプレイテレビ(CZ-601D、CZ-611Dなど)が必要です。※3 モデムユニットCZ-8TM2に同梱のソフトはX1/X1 turboシリーズ用です。※4 使用に際しては、あらかじめ、別売の1MB増設RAMボードCZ-6BE1Aを増設してください。

アートツールと呼びたい「PRO-68K」シリーズソフト。

品名	型番	価格
イージーオペレーションの統合型表計算ソフト BUSINESS PRO-68K	CZ-212BS	標準価格 68,000円
コマンド型リレーショナルデータベース DATA PRO-68K	CZ-220BS	標準価格 58,000円
ワープロ機能を備えたカード型リレーショナルデータベース CARD PRO-68K	CZ-226BS	標準価格 29,800円
FM音源をフルサポートするサウンドエディタ SOUND PRO-68K	CZ-214MS	標準価格 15,800円
マウスを使った簡単操作の楽譜ワープロ MUSIC PRO-68K	CZ-213MS	標準価格 18,800円
AD PCM機能をサポートしたサンプリングエディタ Sampling PRO-68K	CZ-215MS	標準価格 17,800円
オリジナリティを活かせるポップアートツール NEW Print Shop PRO-68K	CZ-221HS	標準価格 19,800円
スクリーンエディタ内蔵の通信ソフト Communication PRO-68K	CZ-223CS	標準価格 19,800円
ソフトウェア開発に役立つCコンパイラ C compiler PRO-68K	CZ-211LS	標準価格 39,800円
24トラックのMIDIマルチレコーディングソフト Musicstudio PRO-68K	CZ-237MS	標準価格 25,800円
MIDI楽器演奏が楽しめるMUSIC PRO68KのMIDI版 MUSIC PRO-68K〔MIDI〕	CZ-247MS	1月発売予定
ソフトウェア開発ツール THE 福袋 V2.0	CZ-224LS	標準価格 9,980円
マルチタスク、リアルタイムオペレーティングシステム OS-9/X68000	CZ-219SS	12月発売予定
本格財務会計ソフトウェア TOP財務会計	CZ-227BS	標準価格200,000円
AI開発ツール AI-68K(Staff LISP/OPS PRO-68K)	CZ-234LS	標準価格188,000円

ゲームソフト	型番	価格
●ツインビー	CZ-217AS	標準価格 7,800円
●アルカノイド	CZ-222AS	標準価格 7,800円
●沙羅曼蛇	CZ-218AS	標準価格 8,800円
●熱血高校ドッジボール部	CZ-232AS	標準価格 7,800円
●フルスロットル	CZ-231AS	標準価格 8,800円

〈パソコン教室開催のお知らせ〉 X68000、MZ-2861のパソコン教室を開催します。くわしくは、下記までお問い合せください。

札幌(011)642-8111・仙台(022)288-8705・東京(03)260-1161・横浜(045)201-6525・名古屋(052)332-2611・大阪(06)222-7655・神戸(078)291-8715・福岡(092)481-2860

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

表紙絵：Matsubaguchi Tadao

■広告目次

Oh!X

C O N T

●特集

〈スタッフ〉
●編集長／前田 徹 ●副編集長／永野 仁 ●編集／植木章夫 石塚康世 高野庸一 ●協力／有田隆也 中森 章 清水和人 後藤貴行 林 一樹 浅野恵造 山村 一 井本 泰 堀内保秀 荻窪 圭 藤原和典 岡本浩一郎 毛内俊行 野中俊一郎 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 倉持亮一 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／手塚喜美子 千野延明

1989 JAN.
1

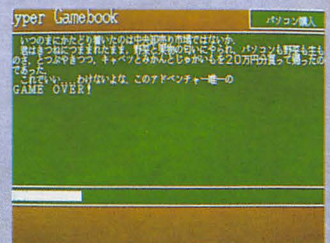
E N T S

●THE SOFTOUCH

●連載/紹介/講座/システム

●読みもの

特集 電子サイコロを作ろう

特集 X68000用CP/M-80システム

Hyper Game Book

Z80マシン語ゲーム工房

Master of Monsters

サンダーフォースII

クリエイティブマインド

21型カラーディスプレイ NEW
CU-21CD
標準価格 139,800円

15型カラーディスプレイ
CU-15M1-E
標準価格 99,800円

RGBシステムチューナー NEW
CZ-6TU-GY・-BK
標準価格 35,800円
(リモコン付)

カラーイメージユニット
CZ-6VT1・-BK
標準価格 69,800円

カラーイメージスキャナ※1
CZ-8NS1
標準価格 188,000円

スキャナ用パラレルボード
CZ-6BN1
標準価格 29,800円

1MB増設RAMボード
(CZ-600C内蔵用)
CZ-6BE1
標準価格 35,000円

1MB増設RAMボード
(CZ-601C/611C内蔵用)
CZ-6BE1A
標準価格 38,000円

2MB増設RAMボード※3
CZ-6BE2
標準価格 79,800円

4MB増設RAMボード※3
CZ-6BE4
標準価格 138,000円

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円

GP-IBボード
CZ-6BG1
標準価格 59,800円

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円

FAXボード NEW
CZ-6BC1
標準価格 79,800円

MIDIボード NEW
CZ-6BM1
標準価格 26,800円

拡張I/Oボックス(4スロット)
CZ-6EB1
標準価格 88,000円

表示

映像入力

画像入力

内蔵メモリ

拡張ボード

拡張スロット

●写真はCZ-601C/CZ-603Dです。

●写真はCZ-611C/CZ-601D/CZ-6ST1です。

●本体+キーボード+マウス・トラックボール

型番	標準価格
CZ-600C-E・-B	標準価格 369,000円
CZ-601C-GY・-BK	標準価格 319,800円
CZ-611C-GY・-BK	標準価格 399,800円

●15型カラーディスプレイテレビ

型番	標準価格
CZ-600D-E・-B	標準価格 129,800円
CZ-601D-GY・-BK	標準価格 119,800円
CZ-611D-GY・-BK	標準価格 145,000円

●14型カラーディスプレイ
CZ-603D-GY・-BK 標準価格 84,800円 NEW
(チルトスタンド同梱)

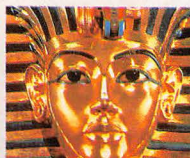
●チルトスタンド
CZ-6ST1-E・-B 標準価格 5,800円
(CZ-600D/601D/611D用)

●高性能CRTフィルター
BF-68PRO 標準価格 19,800円 NEW
(CZ-600D/601D/611D/603D/CU-15M1用)

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1 標準価格29,800円で接続してください。
CZ-6BE1標準価格35,000円(CZ-600C)、CZ-6BE1A 標準価格38,000円(CZ-601C、CZ-611C)を増設してください。

シャープ株式会社 ●お問い合わせは……シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

思わず熱くなる。

あふれる周辺機器がX68000をサポート。

シャープペリフェラルファミリー

X68000

プリントアウト

24ピン漢字プリンタ(80桁)
CZ-8PK7
標準価格122,000円
(信号ケーブル同梱)

24ピン漢字プリンタ(136桁)
CZ-8PK8
標準価格152,000円
(信号ケーブル同梱)

24ピン漢字プリンタ(80桁)
CZ-8PK9
標準価格89,800円
(信号ケーブル同梱)

熱転写カラー漢字プリンタ
CZ-8PC2
標準価格69,800円
(信号ケーブル同梱)

NEW
熱転写カラー漢字プリンタ
CZ-8PC3
標準価格65,800円
(信号ケーブル同梱)

ビデオプリント

カラービデオプリンタ
CZ-6PV1
標準価格198,000円
(信号ケーブル同梱)

ハードディスク

ハードディスクユニット(20MB)
CZ-620H
標準価格178,000円

通信

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格7,200円

モデムユニット※2
CZ-8TM2
標準価格49,800円
(RS-232Cケーブル同梱)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格7,200円

サウンド

アンプ内蔵スピーカーシステム
(2本1組)
AN-160SP
標準価格59,800円

システムラック

システムラック
CZ-6SD1
標準価格44,800円

入力

マウス
CZ-8NM2A
NEW
標準価格6,800円

ジョイカード
CZ-8NJ1
標準価格1,700円

トラックボール
NEW
CZ-8NT1
標準価格13,800円

※2 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。　※3 ご使用に際しては、あらかじめ別売の1MB増設RAMボード

X1・X1turboシリーズ用周辺機器

カラーディスプレイ		
●21型カラーディスプレイ※1	CU-21CD	139,800円
●15型カラーディスプレイ	CU-15M1-E	99,800円

映像・画像入力編集装置		
●カラーイメージスキャナ	CZ-8NS1	188,000円
●カラーイメージボードII	CZ-8BV2	39,800円
●立体映像セット	CZ-8BR1	29,800円
●パーソナルテロッパ※2	CZ-8DT2	44,800円

FM音源		
●ステレオタイプFM音源ボード	CZ-8BS1	23,800円

スピーカー(2本1組)標準装備、ミュージックツール同梱

プリンタ		
●24ピン漢字プリンタ(80桁)	CZ-8PK7	122,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK5	129,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK8	152,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK6	159,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK9	89,800円
●熱転写カラー漢字プリンタ	CZ-8PC3	65,800円
●熱転写カラー漢字プリンタ	CZ-8PC2	69,800円
●カラービデオプリンタ	CZ-6PV1	198,000円

ファイル		
●ミニフロッピーディスクユニット(2HD・2D)※3	CZ-520F	118,000円
●ミニフロッピーディスクユニット(2D)	CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F	19,800円
●ハードディスクユニット※3	CZ-500H	348,000円
●ハードディスクユニット(増設用)	CZ-501H	258,000円
●カセットデータレコーダ	CZ-8RL1	24,800円
●ミニフロッピーディスク	CZ-5M2D/CZ-5M2HD	(各10枚入)
●コンパクトフロッピーディスク	CZ-3FBD	1,300円

拡張ボード・その他		
●モデムユニット(300ボー)	CZ-8TM1	29,800円
●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●ROM BASICボード※5	CZ-8RB	19,800円
●グラフィックRAMボード※6	CZ-8BGR2	14,800円
●RS-232C・マウスボード※7	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※8	CZ-8BF1	14,800円
●JIS第1水準漢字ROM※9	CZ-8BK2	19,800円
●JIS第2水準漢字ROM※10	CZ-8BK4	6,800円
●JIS第2水準漢字ROM&ターボ博士レキシコン・日本語百科ワードパワー※11	CZ-8BK3	13,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張I/Oポート※12	CZ-8EP	11,800円
●拡張I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※13	AN-58C	2,980円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円
●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド※14	CZ-6ST1-B・-E	5,800円
●チルトスタンド※15	CZ-81T-S・-R	8,500円
●高性能CRTフィルター※14	BF-68PRO	19,800円
●システムスタンド	CZ-8SS2	5,500円
●スキャナ用パラレルボード※16	CZ-8BN1	27,800円

(価格は標準価格です。)

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉・S〈メタリックシルバー〉・R〈ローズレッド〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用V1.0 ※6 CZ-850C用 ※7 X1シリーズ用 ※8 CZ-850CでCZ-520Fを使用する場合に必要 ※9 CZ-800C、801C、802C、803C、811C、820C用 ※10 CZ-856C用 ※11 CZ-850C、851C、852C、862C用 ※12 CZ-800C、802C用 ※13 CZ-820C、822C、830C用 ※14 CZ-600D、601D、611D、880D、830D、CU-15M1用 ※15 CZ-801D、802D、811D、850D、855D、870D用 ※16 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

電子機器事業本部テレビ事業部第4商品企画部〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

SHARP

写真のグラフは表示例で、
グラフ作成ソフトは内蔵されていません。

このポケコンが、プロの新しいスタンダードになる。

40桁×4行
プログラム編集に便利なワイド表示。
しかも240×32ドットのフルグラフィック対応。

演算速度7倍
新開発カスタムCPUの採用により、
従来機PC-1475の約1/7の時間で高速処理。

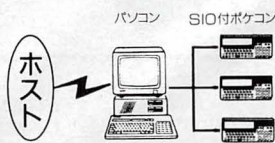
MAX.96KB
別売RAMカードで、ここまで拡張可能。
標準装備：大容量32KバイトRAM。

技術計算に即戦力。エンジニアソフトウェア〈1101機能〉搭載。

●複数のプログラムやデータを本体RAM内で管理できるラムファイル機能●電卓なみの手軽さで関数計算が扱える関数電卓モード●連立方程式もこなせる行列演算機能●入力したデータの確認や修正が簡単にできる統計回帰計算機能●99種までの数式や定数を記憶できる数式記憶機能●有効桁数20桁の高精度演算を可能にする倍精度BASIC搭載●経済的な単4乾電池使用●プログラムやデータの管理に便利なポケットディスク対応●シリアルインターフェイス装備●外形寸法：幅200mm×奥行100mm×厚さ14mm●重量：250g（電池含む）

高機能関数ポケットコンピュータ
PC-E500
標準価格28,800円

Z80*CPU、24桁4行表示
2変数統計機能つき86関数機能
PC-E200
標準価格22,000円
＊Z80はザイログ社の登録商標です。

ポケコンの世界が、いま、どんどん面白くなっている。
POCKET通信Ver.2
㈱工学社のホストの情報が一層充実、しかも本格的なパソコン対応になりました。最寄りのパソコンでアクセスし、必要な情報をポケコンにダウンロード。学校や職場のパソコンがポケコンの生きた情報基地になります。

シャープ株式会社 資料のご請求、お問い合わせは…シャープ㈱コンシューマーセンター OA相談室まで。

東日本OA相談室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161（大代表）
名古屋OA相談室 〒454 名古屋市中川区山王3丁目5番5号 ☎(052)332-2611（大代表）
西日本OA相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221（大代表）
福岡OA相談室 〒816 福岡市博多区井相田2丁目12番1号 ☎(092)575-2381（代表）

資料請求券
PC-E500
oh! X
1係

7号店は、外から見て2FにX68000のシンボル「ツタンカーメン像」がめじるしです。

ツクモ ここ一番のズバリセール

X68000の事なら何でも！
ツクモは「スーパーX PRO SHOP」
7号店2Fはシャープ専門フロアーです。

X68000 ACEシリーズ 好評発売中！

X68000 ACE HD CZ-611C 定価¥399,800➡メーカー希望価格¥383,000
X68000 ACE CZ-601C 定価¥319,800➡メーカー希望価格¥299,000

しっかりものディスプレイ他

- CU-21CD 21型カラーディスプレイ 定価¥139,800 ……メーカー希望価格¥111,000
- CZ-601D ドットピッチ0.39ミリ……定価¥119,800
- CZ-611D ドットピッチ0.31ミリ……定価¥145,000
- CZ-603D ドットピッチ0.31ミリ……定価¥84,800
- CZ-6ST1 チルト台……定価¥5,800
- CZ-6TU RBGシステムチューナー……定価¥35,800
- BF-68PRO 14・15インチCRTフィルター……定価¥19,800

豊富な周辺機器

- CZ-6BE1 1MB増設RAMボード(CZ-600C用)・定価¥35,000
- CZ-6BE1A 1MB増設RAMボード(ACEシリーズ用)……定価¥38,000
- CZ-6BE2 2MB増設RAMボード……定価¥79,800
- CZ-6BE4 4MB増設RAMボード……定価¥138,000
- CZ-6NS1 カラーイメージスキャナー……定価¥188,000
- CZ-6VT1 カラーイメージユニット……定価¥69,800
- CZ-6BP1 数値演算プロセッサボード……定価¥79,800
- CZ-6BC1 FAXボード……定価¥79,800

よくできたソフトはよいお店で

- New Print Shop PRO-68K ポップアートツール……定価¥19,800
- CARD PRO-68K カード型リレーショナルデータベース……定価¥29,800
- MUSIC PRO-68K 楽譜ワープロ……定価¥18,800
- SOUND PRO-68K サウンドエディタ……定価¥15,800
- Sampling PRO-68K サンプリングエディタ……定価¥17,800
- Z's STAFF PRO-68K プロ仕様グラフィックエディタ……特価¥49,800
- C Compiler PRO-68K Cコンパイラ……定価¥39,800
- OS-9/X68000 X68000用OS9 係員にお尋ね下さい。
- C & Proffesional Package OS-9用Cコンパイラ……係員にお尋ね下さい。

＊その他、ビジネスソフト・ホビーソフトも多数発売中ですのでお気軽にお尋ね下さい。

ツクモでハードディスク

お財布にやさしい価格！

●ウインテク
- HD-202(20MB 85ms)……ツクモ特価¥53,800
- HD-404HS(40MB 28ms)……ツクモ特価¥103,800

●アイテック
- IT X-203(20MB 28ms)……ツクモ特価¥73,800
- IT X-403(40MB 29ms)……ツクモ特価¥99,800

(X-203/403はブラックかグレーをご指定下さい。)

MIDIは最高！

唄うのが下手でもピアノが弾けなくても関係ないョ！

- CZ-6BM1 MIDIボード……定価¥26,800
- CZ-237MS Music Studio PRO-68K・定価¥25,800
- MT-32 ローランドMIDIサウンドモジュール・定価¥69,000

♬セットで揃えるととってもお安くなります。お問い合せ下さい。

驚異の大特価！NEW Z-BASIC

X1ターホシリーズ対応 CZ-141SF (32KBメモリーホート付属)
ツクモ特価¥9,800 (〒1.000)

新登場！ X1 turbo Z III

- ●CZ-888C-BK……¥169,800
- ●CZ-860D-BK……¥99,800

セット特価販売中

★上記セットに買い換えるなら

下取り機種	差額
CZ-852C＋CZ-850で	¥181,000
CZ-856C＋CZ-870で	¥175,000
CZ-822C＋CZ-820で	¥197,000

モデム

オムロン MD-2400B
300／1200／2400ボー
ツクモ特価¥39,800

通信ソフト
X1 turbo用
SPS JETターボターミナル……特価¥8,400
X68000用
SPS た〜みのる……特価¥10,900
シャープ Communication PRO-68K…標準価格¥19,800

プリンター

- CZ-8PC2 カラー熱転写プリンター……ツクモ特価¥49,800
- CZ-8PC3 カラー漢字熱転写プリンター……メーカー希望価格¥61,800
- CZ-8PK6 24ピン漢字ドットプリンタ(15インチ)……ツクモ特価¥69,800
- MZ-1P17 ケーブル、第2水準ROM付……ツクモ特価¥39,800

X1G モデル30セット

- ●CZ-822CB 本体
- ●CZ-820DB ディスプレイ
- ●人気ゲームソフト
- ●オリジナルゲームパック
- ●ディスケット……サービス

ツクモ特価 ¥79,800

※X1 twinも特別販売中ですョ！

マウス／トラックボール

(X1、X1turboシリーズ／MZ-2500シリーズ対応)

ツクモオリジナルマウスセット
TS-MX1＋マウスパッド ツクモ特価¥5,500
シャープトラックボール
CZ-8NT1…定価¥13,800

ポケコン／電子手帳

シャープ PA-8500
定価¥28,000
大型4行表示、データスケジュール管理に便利。ICカード、プリンタで更に発展するハイグレードタイプ。
特価 ¥24,800

シャープ PC-E200
定価¥22,000
特価¥17,800

シャープ PC-E500
定価¥28,800
特価¥24,800

営 AM10時〜PM7時
年内無休、年始は2日より営業！

全国代金引き換え配達
お申し込みは☎03-251-9911へお電話1本！
商品到着の際、玄関でお会計ができます。配達日の指定もできます。

夏・冬、ボーナス2回払い受付中
月々¥3,000以上の均等払いも頭金なし。

現金書留なら
〒101-91 東京都千代田区神田郵便局私書箱135号 九十九電機㈱通信販売部

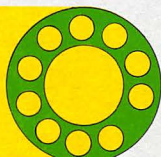
銀行振込なら
事前に☎でお届け先をご連絡下さい。
富士銀行 神田支店 普No.894047

PRO STAFF ツクモ

九十九電機(株) 〒101-91 東京都千代田区神田郵便局私書箱135号

通信販売部	03-251-9911
ツクモ7号店	03-253-4199
ツクモ5号店	03-251-0531
ニューセンター店	03-251-0987
名古屋1号店	052-263-1655
名古屋2号店	052-251-3399
ツクモ札幌	011-241-2299

案ずるより買うが易し。

あれこれ迷っている人、すぐウェーブ・アイへTELして下さい。

お申し込みは料金無料のフリーダイヤルで

0120-009898

だから安心 ウェーブ・アイ10ポイントチェック

チェック1 全品2倍保証！
メーカー保証の2倍の保証がついています。(メーカー1年保証＋ウェーブ・アイ1年保証)

チェック2 冬のボーナス一括払いOK！
商品は今すぐお手元に、お支払いはまとめて冬のボーナスで!!

チェック3 超低金利クレジット
3回〜72回までのクレジットが格安の金利でOK。また当社提示支払い例のほかにお客様独自の支払いプランが組めます。

チェック4 商品先取り、支払いは半年先から。
支払い開始は半年先！でも商品はほしいというお客様でもOK！

チェック5 ボーナス2回払いOK！
月々の支払いは全くナシ！お支払いは冬と夏のボーナスでOK！

チェック6 代金引換OK！
現金一括にしたいというお客様、お支払いは現品到着時でOK！(但し離島の方はご利用できません。)

チェック7 全国無料配送
一部地域を除き送料無料で商品をおとどけします。(但し5万円以上の商品に限ります。離島の方は有料となります。)

チェック8 配達日指定OK！
留守がちの方の為に、ご都合に合わせて、配達致します。もちろん日曜・祭日もOK。

チェック9 下取り、買取りもOK!
お手持ちのパソコンを下取りして、わずかな予算で新製品と買い換えることができます。

チェック10 ハガキ注文もOK！
いそがしくて電話をするひまが無いという方の為に、ハガキでのご注文もOK!

〒252 藤沢市湘南台一丁目一〇番地一号 ウェーブアイ Oh!X係

1. 住所
2. 氏名
3. 年令
4. 電話番号
5. 保護者名 印 (20才未満の方)
6. 商品名
7. 支払い方法
   月々 円× 回
   ボーナス 円× 回

X68000 ACE HD

X68000に20MBハードディスクを搭載。ますます熱くなるクリエイティブ＆パーソナルワークステーション。

プラン150 X68000ACE HD お買得基本セット TELにて

品目	価格
CZ-611C	399,800円
CZ-603D(0.31ミリ、高解像度CRT)	79,800円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	502,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
7,000円×36回	ボーナス27,300円×6回
5,000円×48回	ボーナス23,800円×8回
3,000円×60回	ボーナス26,800円×10回
6,400円×72回	ボーナス なし

プラン151 X68000ACE HD 純正基本セット TELにて

品目	価格
CZ-611C	399,800円
CZ-601D(0.39mm、TV内蔵CRT)	119,800円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	542,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
8,000円×36回	ボーナス26,900円×6回
5,000円×48回	ボーナス28,200円×8回
4,000円×60回	ボーナス24,500円×10回
6,900円×72回	ボーナス なし

プラン152 X68000ACE HD アートセット TELにて

品目	価格
CZ-611C	399,800円
CZ-603D(0.31ミリ、高解像度CRT)	79,800円
CZ-8PC3(80桁、カラー熱転写プリンター)	65,800円
Z's STAFF PRO-68K(グラフィックツール)	58,000円
PRINT-SHOP-PRO68K(高性能ポップアートシール)	19,800円
CZ-6TU(RGBシステムチューナ)	35,800円
CZ-6VT1(カラーイメージユニット)	69,800円
A4カット紙100枚	480円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	752,280円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
12,000円×36回	ボーナス40,800円×6回
10,000円×48回	ボーナス27,600円×8回
8,000円×60回	ボーナス25,000円×10回
10,500円×72回	ボーナス なし

プラン153 X68000ACE HD ミュージックセット TELにて

品目	価格
CZ-611C	399,800円
CZ-601D(0.39mm、TV内蔵CRT)	119,800円
CZ-8PC3(80桁、カラー熱転写プリンター)	65,800円
MUSIC PRO-68K(楽譜ワープロ)	18,800円
SOUND PRO-68K(FM音源サウンドエディタ)	15,800円
ED-700(2段パソコンラック)	27,000円
A4カット紙100枚	480円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	670,480円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
12,000円×36回	ボーナス26,800円×6回
9,000円×48回	ボーナス22,700円×8回
7,000円×60回	ボーナス22,000円×10回
9,200円×72回	ボーナス なし

X68000 ACE

ハードの余裕がフレンドリーなオペレーションを生みだしている。ますます熱くなるクリエイティブワークステーション。

プラン147 X68000ACE純正お買得基本セット TELにて

品目	価格
CZ-601C	319,800円
CZ-603D(0.31ミリ、高解像度CRT)	79,800円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	422,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
10,000円×24回	ボーナス24,800円×4回
6,000円×36回	ボーナス22,300円×6回
4,000円×48回	ボーナス21,300円×8回
6,200円×60回	ボーナス なし

プラン148 X68000ACE純正基本セット TELにて

品目	価格
CZ-601C	319,800円
CZ-601D(0.39mm、TV内蔵CRT)	119,800円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	462,600円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
6,000円×36回	ボーナス26,100円×6回
4,000円×48回	ボーナス24,200円×8回
3,000円×60回	ボーナス22,200円×10回
5,700円×72回	ボーナス なし

プラン149 X68000ACEミュージックセット TELにて

品目	価格
CZ-601C	319,800円
CZ-603D(0.31ミリ、高解像度CRT)	79,800円
CZ-8PC3(80桁、カラー熱転写プリンタ)	65,800円
Music PRO68K(簡単操作の楽譜ワープロ)	18,800円
Sound PRO68K(FM音源サウンドエディタ)	15,800円
Sanpling PRO 68K(高機能サンプリングエディタ)	17,800円
AN-160SP(アンプ内蔵スピーカーシステム)	59,800円
A4カット紙100枚	480円
ブランクディスケット3M(5インチ2HD)10枚	23,000円
定価合計	601,080円

クリーニングディスク・マウスパッドサービス

ウェーブ・アイ特価

月々	ボーナス
10,000円×36回	ボーナス29,400円×6回
7,000円×48回	ボーナス27,400円×8回
5,000円×60回	ボーナス27,900円×10回
8,300円×72回	ボーナス なし

冬の 商品今すぐ!! ボーナス一括払いOK

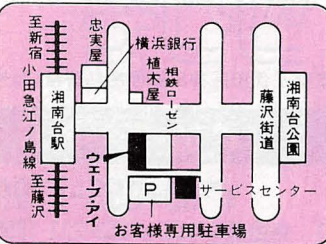
受付時間 AM9:30▶PM9:00 電話一本即納！

店	電話	店	電話
藤沢	0466(43)1775	静岡	0542(54)0696
札幌	011(771)4971	名古屋	052(581)4325
盛岡	0196(24)3172	大阪	06(362)5057
仙台	022(267)5371	広島	082(293)0811
新潟	0252(75)5076	福岡	092(481)0502
東京	03(226)9286	FAX	0466(43)1265

18歳未満の方は保護者とご一緒にお電話下さい。

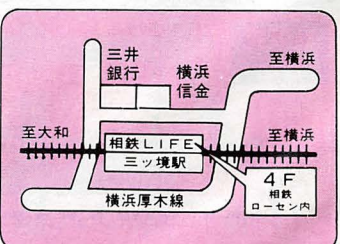
新歩人の情報ターミナル

湘南台店☎0466-43-1771

未来をクリエイトする WAVE EYE ウェーブ・アイ

〒252神奈川県藤沢市湘南台1-10-1

振込銀行▶横浜銀行 湘南台支店 当座000467 (株)ウェーブ・アイ

第二・第三火曜日定休日

三井銀行 横浜信金 至横浜 至大和 相鉄LIFE 三ッ境駅 至横浜 横浜厚木線 4F 相鉄ローゼン内

三ッ境店☎045-363-7044

OS-9/X68000

Real-Time Multi-Tasking Operating System

漢字バージョン V2.2 Release 1

プログラミング・レベルからアプリケーション・レベルまで

使いやすく機能的な環境を作り出します。

プロ指向EWSレベルの
マルチタスク・マルチウインドウ
　メモリの許すかぎり、いくつでも
　ウインドウを開くことができます。

AD-PCM、FM音源と
グラフィックスをマルチタスク
処理可能

AV機能標準サポート

不思議ソフトAVRIDER標準

グラフィックス・サポート
　768×512その他

Human 68Kとのファイル互換性

日本語処理標準装備
　漢字辞書はHuman68Kと共通

マルチタスク
リアルタイムI／Oサポート

ファイル名、プログラム名の
漢字サポート

PSS(プレゼンテーション・サポート
システム)によるグラフィックス・
マウス・サポート

VRAM DISK、SRAM DISKサポート

ダイレクトTVコントロール

オート・パワーon／offコントロール

販売元：シャープ株式会社

シャープ㈱電子機器事業本部システム機器営業部　〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部　〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

開発元：マイクロウェア・ジャパン株式会社

近日発売

マイクロウェア・オリジナル・ツール・パッケージ　(OS-9/X68000本体はシャープ㈱よりお買い求め下さい。)

C&プロフェッショナル・パッケージ
OS-9/X68000のプログラム開発を強力に支援するパッケージです。

プログラマーズ・ツール・キット
OS-9/X68000のドライバ開発を強力に支援するパッケージです。

OS-9/X68000 SRCDBG
OS-9／X68000用Cコンパイラに完全に準拠したソース・レベル・デバッガです。

OS-9/X68000ネットワーク・システム
ローカル・エリア・ネットワークを構築するためのハード&ソフト・パッケージです。

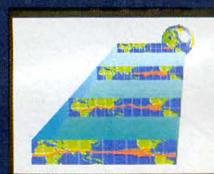
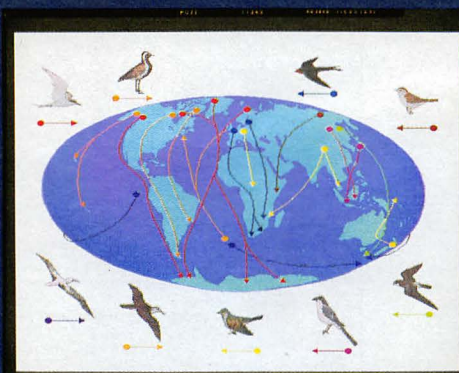
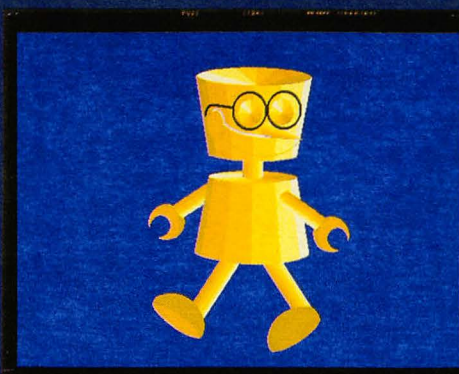
マイクロウェア・ジャパン株式会社　〒273 千葉県船橋市本町4-41-19 本町セントラルビル7F PHONE：0474-22-1747

更に、先へ。

アーティスト達の現場の貴重な声に応えて、新しくバージョンアップしたシーズスタッフPRO-68K［Ver.2.0］。注目のアウトラインフォント搭載をはじめ、多彩な入出力機能を装備。それはまさに、プロスペックを超えたプロスペック。親愛なるユーザーへ。いま、ツァイトから。

内田美智子 ―――― リベラルペインター
●ポスター・文庫本・カット用イラスト。
他にオブジェ、内装、パフォーマンス、ヘアーメイクなど
活動範囲は多岐にわたる。
Photograph:透明色利用、グラデーション・マスク機能により、
その表情をより豊かに。

倉嶋正彦 ―――― ビジュアリスト
●CGイラストレーションのTV番組用タイトル作成。
NTV「巨泉のこんなモノいらない!?」オープニング及びデータCG。
フジテレビ「オート倶楽部」オープニング、
「プロ野球ニュース」コーナー、「トークシャワー」オープニング、
テレビ東京「パソコンサンデー」オープニングなどで活躍中。
Photograph left:「トークシャワー」
ロゴのカラーシミュレーション。
Photograph right:「巨泉のこんなモノいらない!?」
テーマのイメージCG。

宮嶋美奈子 ―――― CGイラストレーター/中央工学校兼任教師
●IMAGICAのキャプテンシステム用画像データデザイン、
福武書店のSTUDY BOX用画像データ、
ノーリツのショールーム(NOVANO)浴室パースデータ、
ハイテックラボの「CD-ROM ON CD-ROM」用画像データ、
各種ゲームソフト用画像データなどを制作。
Photograph:C-TRACEで作成した地球を
PRO-68Kに転送し、キャラクターと合成。

長谷川一光 ―――― イラストレーター
●著書「たまごから生まれた赤ちゃん」、
LION/チャーミークリーンなどの植物画から
森永乳業・セブンイレブンのメルヘンタッチのイラストなどを制作。
Photograph left:「渡り鳥地図」スキャナで原画を取り込み、
原画に忠実な色を調合、そのリアリティを高めた。(JAF-MATEに掲載)
Photograph right:「エルチチョン火山の噴煙」
(学研5年の科学に掲載)

喜多見康 ―――― イラストレーター
●雑誌「ホットドックプレス」・「トレンディ」の表紙、
雪印乳業のアイスクリーム用ポスター、
名古屋駅前ファッションビル・アピタのクリスマス用ポスターなど。
Photograph left:ユニークなキャラクターを次々に制作中。
Photograph right:イメージとしての花をビジュアル化。

プロスペックを超えたプロスペック。

ジーズスタッフPRO-68K[Ver.2.0]近日発売。

ビギナーから使える、プロスペックソフトです。

65,536色という圧倒的な表現力をサポートし、透明色描画、各種グラデーション、マスク機能、さらにイメージユニット、スキャナなど。高度な応用グラフィックスに自在に対応。しかも注目のアウトラインフォント(JIS第一水準の明朝&ゴシック)を搭載。操作性の面では、最大16枚ものウィンドウを同時表示・同時機能させる他、縦2画面分512×1024ドットのスクロール編集が可能。プロスペックをぬり変える、新しいプロスペック—— PRO-68K[Ver.2.0]の誕生です。

パレット:ジーズスタッフPRO-68K[Ver.2.0]ではX68000のもつ"65,536色同時発色"というこれまでにないハイスペックを存分に生かすために、色の調合に次の方法を設けています。32階調の透明色利用、グラデーションボードによる中間色調合、HSV調合、さらに基本となる384色を用意したカラーセレクトです。もちろんそれぞれは目的に応じて、容易に使い分けることができます。■HSV:色相、明度、彩度の調節により微妙な色が作れます。■タイル:16種類(16×16ドット)タイルパターンを絵の具の代わりに利用できます。■タイルエディット:グラデーションボード、32階調濃淡設定、4倍トーン編集、登録が行なえます。

ペン:ジーズスタッフPRO-68K[Ver.2.0]ではその名に恥じない、まさにプロスペックなペンを豊富に用意しています。しかも必要に応じてオリジナルのペンも作成可能。太さは3種類(4×4・8×8・16×16ドット)でペン先は18種類。それぞれ32階調の透明度設定を用意。また半径、密度、32階調の濃淡設定、エアブラシ、コンパス、楕円コンパス、スプライン、スムース、コネクト、ボックス、ボックスフィル、クローズドペイント、マスク/アンマスク/マスククリア/マスクリバースなど、プロユースにふさわしい装備を施しています。■ペンエディット:任意のペン先が作れ、登録できます。■エアブラシ:濃淡、直径の設定はもちろん編集が可能。■ペイント境界領域設定:色相、明度、彩度の設定により境界領域に指定ができます。

エディット:コピーやムーブはボックス、任意閉曲線、コピーカラーの設定が行なえ、他にも拡大/縮小、上下/左右反転、回転、カラーチェンジ、トランスフォーム、ルーペ、特殊機能/モザイク/フォーカスなど多彩な機能を装備しています。■コピーカラー:グラフィックスの中から必要な部分だけを取り出しコピーすることができます。■ルーペ:任意の箇所を2/4/8/16倍まで拡大・修正できます。■カラーチェンジ:HSV機能により、自然な感じでカラー変換が行なえます。

文字:漢字変換はもちろん、字色・縁色・影色をそれぞれ指定可能。他にも斜体が作れ、また文字サイズは24ドットを標準とし、12ドットから192ドットまで12単位で縮小・拡大が行なえます。さらにアンダーライン(コネクト・スプライン・サークル)にそって文字入力が可能。イメージどおりの文字処理が手軽に行なえます。■アウトラインフォント:JIS第一水準の明朝&ゴシックを搭載しています。

数値:長さや角度、比率などのリアルタイム表示が、細密な描画をサポートします。

オプション:豊富なオプション機能を用意しています。■イメージスキャナ:手持ちの原画・写真が手軽に入力できます。■イメージユニット:カメラ入力の画像処理が行なえます。■イメージレイアウト:縦2画面分のレイアウトが縮小されて表示されます。全体イメージの確認が容易に行なえます。■コントロール:ダブルクリック時のクリック間隔を調節できます。■メッシュ:画面に方眼をかけることができます。間隔の設定が行なえます。■アンドウ:ミス操作のやり直しができます。

入力機器●マウス●イメージスキャナ:NEC IN-501/502/503(16階調ハーフトーン)、EPSON GT-3000/3000V/4000、SHARP JX-200、OMRON HS-10R/別売りイメージユニットに完全対応です。イメージユニットを介することで、ビデオカメラによる入力が行なえます。

出力機器●A4縦、またはハガキサイズにハードコピーが可能。●プリンタ:NEC PC-PR201/CL/T/TL/H/H2/HC/V/F/PR801/PR101/T/TL/F/L/PR406/NM-9900/9950、横河電機NP510、SHARP IO-725/730、EPSON ESC/P24-J83C対応機種。また●ビデオ出力が行なえる他、ビデオプリンタとの接続により鮮やかなプリントアウトが可能。●ポジフィルム:フジカラーサービス(五反田)プロフェッショナル・フォト課CG係

PROFESSIONAL GRAPHIC SOFT

Z'S STAFF PRO-68K [Ver.2.0]

ジーズスタッフPRO-68K[Ver.2.0] ¥58,000 特製ハードケース入り

無償バージョンアップセールを実施中。

◉12月10日~ジーズスタッフPRO-68K[Ver.2.0]の発売日までに、無償バージョンアップ交換シールの付いたPRO-68Kをお求めの方に限り、無償でバージョンアップを実施しています。詳しい内容はシールおよび店頭でご確認ください。◉バージョンアップサービスを実施いたします。ジーズスタッフPRO-68Kのユーザー登録がお済みでない方は、ユーザー登録カードを至急お送りください。バージョンアップサービスのご案内をお送りいたします。

zeit 株式会社ツァイト

〒151 東京都渋谷区初台1-47-1 小田急西新宿ビル
ユーザーサポート係 ☎03-299-0461

Human Hudson

パワーリーグにシーズン・オフはない。

パワーリーグ

©1988 HUDSON SOFT

かつてないリアリティで迫る X68000版 12月新登場! 希望小売価格¥7,800

「アウト!」「セーフ!」高橋名人の声がジャッジ!

プロ野球も終わって、ポッカリ気の抜けた野球ファンよ。ガッカリしている場合ではない。あのパワーリーグが、X68000初の本格派ベースボールゲームとして、いよいよ新登場する。ボールはカーブ、シュート、ホームランなど遠近感や立体感は野球中継を超えたリアリティ。X68000ならではの鮮明画像が、かつてない迫力の野球ゲームを実現した。その上、サンプリング機能により高橋名人の生のヴォイスが「アウト」「セーフ」をジャッジする。しかも、選手のデータを見ながら、チームの編成、選手の交代まで監督としての采配も存分にふるえるときている。オールシーズン、パワーリーグにオフはない。

さあ、始まりました世紀の対決、いよいよゲーム開始です。

1アウト、ランナー2・3塁、はやくも絶好のチャンスをむかえています。

バックホーム体勢ですか。トップアイから外野守備のフォーメーションを見てみましょう。

試合終了。ゲームの経過をニュースでふりかえってみたいと思います。

X68000でのパワーリーグのプレイには、ジョイカードスーパーXが必要です。(従来のジョイカード/スーパーIIも使えます)

希望小売価格 ¥2,200

本　社　〒062 札幌市豊平区平岸3条5丁目1番18号 ハドソンビル TEL 011-841-4622
東京支社　〒162 東京都新宿区市谷田町3丁目1番1号 ハドソンビル TEL 03-260-4622
大阪支店　〒542 大阪市南区鰻谷仲之町57番地 大阪料理会館ビル5階 TEL 06-251-4622
営業所　札幌・東北・名古屋・福岡

「C-TRACE 68」は、

演算スピードが3倍になりました!!

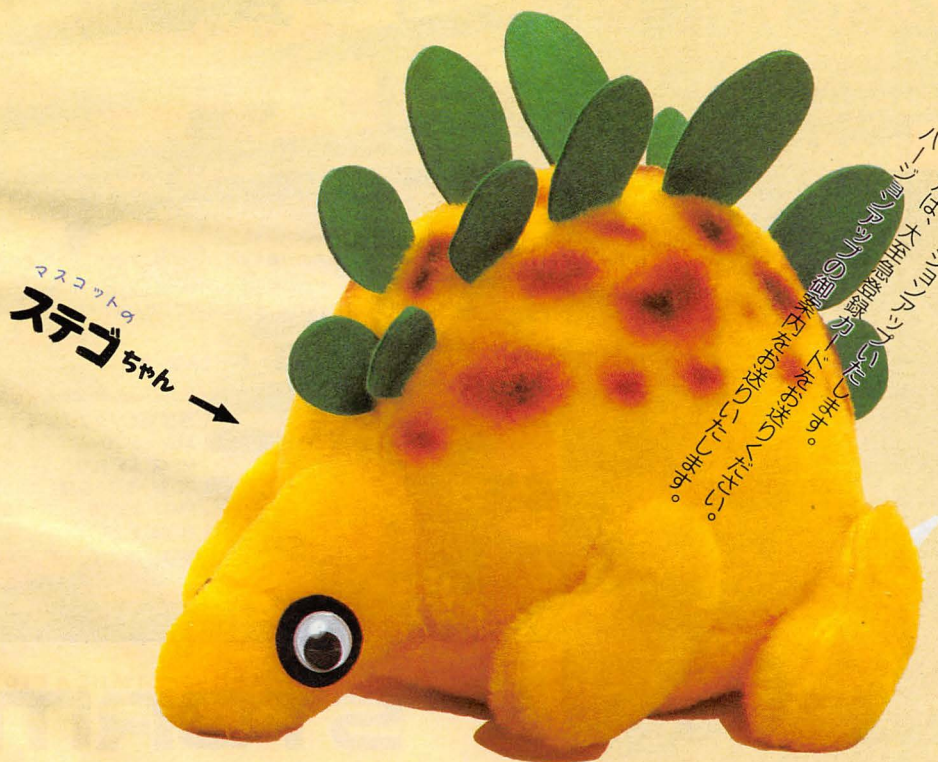
（数値演算プロセッサボード着装時）

3次元コンピュータグラフィックス

レイトレーシングソフトウェア

C-TRACE 98 DRY（全機種対応） ¥68,000
C-TRACE 68（X68000対応） ¥68,000

近日発売 C-TRACE 98＋（全機種対応） ¥198,000
C-TRACE NEWS（SONY） ¥380,000

■動作環境
PC-9801シリーズ全機種
RAM 640KB
MS-DOS Ver.2.11以上
コプロセッサー（8087、80287、80387）有無どちらも対応
■現在サポートしているフレームバッファ（X68000は本体のみ）
PC-9801 本体内VRAM
写像 SIG社／スーパーフレーム サピエンス社
ハイパーフレーム デジタルアーツ社

▶商品のご注文は、お近くのマイコンショップまたは現金書留、郵便為替にて、商品名・機種・メディア名を明記の上、弊社までお申込み下さい。

Cast 株式会社 キャスト
〒158 東京都世田谷区等々力2-1-13 TEL.03-705-0656 FAX.03-705-5224

C-TRACE98DRYユーザーには、98＋を差額（¥130,000）にて交換いたします。

Micronet CO., Ltd. 〒064 札幌市中央区南10条西15丁目ムラカミビル3F
☎011-561-1370

極道陣取り

ここは某年・某月・某国・某市、何時とも何処ともわからない街、きみはいきなり極道者で組の若い者にも人望があつい。そして某日、極道をきわめた大親分の部屋に呼びだされ、とつぜんこんなことを言われる。「わしは､長くない」きみは、だまっている「どーだおめー、この組をまかされてくれね～か」「おやっさん！なにいってんすか」と詰め寄るきみのほうを見つめゆっくりとうなずくとがっくりと力が抜けた！。頭の人望でもっていた組だ、離れて行くヤツもいる。

今きみは、いくつかのシマを基に組をまとめ、勢力を拡大するのが、親っさんへの恩返しだ！
12都市での抗争を、再現！極道も、殺し屋も、チンピラも、きみの部屋に大集合！きみしだいでこの街は、きみの物だ！「シミュレーションゲーム!?」と聞いただけで、ケイエンしてしまう君。操作がむずかしい、時間がかかる、グラフィックが貧弱、音楽性に欠けるなどなど、いままでのS.L.G（シミュレイションゲーム）に不満をもっていたきみ。「S.L.Gならば、まかせて！」というきみにも、マイクロネットが自信をもってお送りします。

○フルマウスオペレーションで操作ラクラク！
○かくしコマンドもあって、ちょっとしたゲームをするとそれも、おしえてもらえちゃう！
○ヒントはマニュアルのな・か・に？

X1 5"2D……6,800YEN 発売中！
PC-88 5"2D……6,800YEN 12月中旬
PC-98 (5/3.5各種)……6,800YEN 開発中
MSX2 3.5"2DD……6,800YEN 12月下旬

普通に、つかえるスーパーツール

GRAPHIC TOOL
がばん **GABAN**

マウス対応 メモリー 64K VRAM 128K
MSX2 3.5"2DD……9,800yen
他社MSX2用作画ソフトで、製作した作画DATAも読み込むことが可能！

各機種ゾクゾク
麻雀狂時代 SPECIAL

バスマウス対応 2ドライブ専用 256K以上
PC-9801 5"2DD(二枚組)……6,800yen
PC-9801 3.5"2DD(二枚組)……6,800yen
バスマウス対応 256K以上
PC-9801 5"2HD……6,800yen
PC-9801 3.5"2HD……6,800yen
インテリジェントマウス対応 1ドライブ可
FM77AV 3.5"2D……6,800yen

新登場!!

移植版情報！

NEO STRATEGIC SIMULATION
STORM

PC-9801 5"2HD……7,800YEN 12月下旬
PC-9801 3.5"2HD……7,800YEN 12月下旬
PC-9801 5"2DD……7,800YEN 12月上旬
PC-9801 3.5"2DD……7,800YEN 12月上旬

What! OH BUSINESSから秋の新作2本!…

▶Easy Graphic Tool
あなたのイメージをかたちにするのがグラフィックツール

超多機能でも、つかいこなせないツールたち……………………
機能は小さくてもいいのです。つかいやすければ……………
G68Kのいのちはし・な・や・か・さです。

なぜ、G68Kなのか、理由は5つある…………………………

① シンプル操作がとても自然
② 缶ビールを飲みながら……の感覚で操作OK。
③ しかも、低価格だから、快適 ¥14,800
④ マニュアルレス感覚のグラフィックツール
⑤ 美しいサンプル画面データを収録(65536色)

定価 ¥14,800

■G68K機能一覧
●にじみ表現が可能なペン●エアブラシ●直線を引く●長方形を塗りつぶす●拡大・縮少●左右反転●上下反転●複写●塗りつぶし●2つの色を混ぜ合わせ新しい色を作る●イメージスキャナ(GT-3000)をサポート●内山亜紀先生の緻密で綺麗なイラストデータ入り●作業中のBGM付きグラフィックツール (選曲可能)

●Z's STAFF PRO68Kのデータをロードセーブ
●アートマスター400(9801)からZ's STAFF PRO 68Kへのデータコンバート機能
●アートマスター400(PC-9801用)のデータをロード

▶スプライトエディタ E68K

簡単にできる貴方だけのオリジナルグラディグス

定価 ¥19,800

●65536色をサポート●
1つのスプライトに65536色中16色を選択して、1ドット単位で色が付けられます。

●1画面上で64パターンを同時編集●
1画面で64パターンのスプライトデータを編集できます。
(1パターン 16*16ドット)
(ページ切り換え機能により28ページまでメモリーに保存できます)

●アニメーション機能を搭載●
作成したスプライトパターンを8コマまで設定し、動きを決めるとアニメーションできます。(作成したスプライトの動きがすぐに確認できます)

●拡大モードは4種類●
2・4・8・16倍で拡大エディットできます。

●強力な編集機能●
LINE・BOX・BOX FILL・PSETをサポートしています。

●BGM機能●
スプライトエディタでは初のBGM機能搭載。(5曲の中から選曲可能)

●スプライトデータならどんな形式でもエディット可能●
ディスク・メモリーからのスプライトデータの読み込みが可能です。

●増設RAM・ハードディスクをサポート●
増設RAMを接続していると1度にエディットできるデータ量が増えます。
ハードディスクからの立ち上げ、ハードディスクからのデータ読み込みもOKです。

販売代理店:近畿システムサービス㈱

OH! BUSINESS
京都市山科区音羽西林町2 TEL:075-502-2972

The SUPER LAS VEGAS

さあ、いよいよカジノへ乗り込むぞ。

ひとくせ、ありそうな5人が相手だ。

美人ギャンブラーの微笑みの裏には。

ラスベガスの夜景が、ムードを盛り上げる。

緊張のゲームスタートだ。落ち着け！

勝負！勝たなければ後がない!!

その一瞬、ラスベガスへ。君はカジノの支配者になれるか。

ラスベガスのカジノで、つわものギャンブラーたちと虚々実々の駆け引きをしながらの大勝負。ゲームのおもしろさ＋カジノのスリリングなムード、それが「ザ・スーパーラスベガス」です。プレイする楽しさはもちろん、ディスプレイに映し出される鮮明なグラフィックにもご注目！ラスベガスの美しい夜景、カジノの華やかなネオンライト、悲喜こもごもに表情をかえるキャラクターたち…と、トップクラスの鮮やかでキメ細かな画像をお楽しみください。

主人公のカジノ荒らしに扮する君は、勝負に勝って稼がなければ、次のエリアの新たなゲームに挑むことができません。勝ち進んで億万長者になるか、それとも無一文で寂しくカジノを去るか。それは、君の実力とツキしだい。さあラスベガスで、ギャンブラーたちとの勝負にチャレンジだ。

〈ゲーム内容〉マニアからファミリーまで楽しめる、10種類もの多彩なゲームが充実しています。
コントラクトブリッジ/ブラックジャック/セブンスタッドポーカー/スロットポーカー/大富豪/ナポレオン/七並べ/セブンブリッジ/ページワン/ルーレット

●「ザ・スーパーラスベガス」をさらに楽しんでいただくために、コントラクトブリッジなどのプレイ方法をわかりやすく説明した豪華カラー解説書(約160P)が付いています。

11月25日新発売

ザ・スーパーラスベガス
ND-22FD 12,800円
PC-9801シリーズ/X68000用

日本デクスタ株式会社 〒101 東京都千代田区外神田2-9-3 ユニオンビル花家3F ☎03(255)9761(代)

日本デクスタのソフトウェアは、全国の有名パソコンショップでお求めください。また、通信販売で直接オーダーされる際は、現金書留にて日本デクスタ宛お申し込みください。

熱くなる。

X68000 ¥7,800
X1 ¥6,800

プロフェッショナル

麻雀悟空

相手が強いとこんなに面白い。

独自の思考アルゴリズムによりコンピュータは格段の強さ
対戦相手は個性溢れる24人
段位戦・勝抜戦の成績に応じて段位が上がる昇段システム
きめ細かなルール設定が可能
記録室では個人の成績やいろいろなデータが見られます
オープン・リトライ・指導など研究のための機能も充実

※画面写真は全てX68000版のものです

企画・制作―株式会社シャノアール TEL03(702)0851 〒158 東京都世田谷区深沢5-15-15 ラフィーネ015

Falcom
迫り来る“魔”の波動を感じたか!?
ならば旅立て、
暗黒のギルバレス島へ!!
SORCERIAN SYSTEM
SCENARIO
Vol.3
ピラミッド ソーサリアン
Sorcerian System
『戦国ソーサリアン』で暗黒神・邪鬼を倒すことは、“魔”の物語の序章にしかすぎなかった! 暗黒神・邪鬼を凌ぐ強大な“魔”の本体、大魔王ギルバレスに挑む冒険だから、『ピラミッド・ソーサリアン』が、『戦国ソーサリアン』以上のボリュームを持っているのは、当然かもしれないが、それにしてもすさまじい。5インチ2Dディスクでは、やっとのことで2枚に収めたものの、内容の濃さは5枚分!? 1から5までが連なった長編シナリオであるというだけでなく、謎解きあり、アクションありで、各シナリオごとに、その性格が変えられているんだ。これほど彩り豊かで、これほど手ごわいゲームなんて、『ソーサリアン』にもなかったぜ!
SORCERIAN SYSTEM SCENARIO Vol.3
Pyramid Sorcerian
ピラミッド ソーサリアン
12月23日発売!
X1turbo 5.2D(2枚組)3,800円
Falcom 日本ファルコム株式会社
Personal Computer Software
〒190 東京都立川市柴崎町2-1-4 トミオービル
通信販売(送料無料)
●現金書留の場合
氏名・機種名・住所・氏名・電話番号を明記して、現金書留でお申し込みください。
●代金引換の場合
電話やFAXやハガキで、品名・機種名・住所・氏名・年齢・電話番号を明記して、お申し込みください。商品お届け時に商品代金をお支払いください。
TEL 0425(27)6501
FAX 0425(28)2714
SORCERIAN
Sorcerian System
SCENARIO Vol 2
好評発売中!!
SORCERIAN SYSTEM
SCENARIO Vol.2
戦国ソーサリアン

あの名作ディスクミステリーがX68000に甦える

コマンド入力だから面白い、本格派のためのアドベンチャーゲーム

「人生とは素晴らしい舞台、時間はかくも
巧妙な伏線を張った物語を用意して、
我々を楽しませてくれるではないか。」
………クルツフォン・エルガー博士

殺人事件さえもこの「カサブランカに愛を」では一つの伏線にしか過ぎない。殺人事件が解決した後も物語は続く。むしろ、事件解決後こそが物語のメインテーマを語る部分なのだ。時間を超えた壮大な人間ドラマの中にプレイヤーは人を思いやる気持ちの大切さ、生きることの素晴らしさを感じるのである。僕はゲームを解き終わった時、感動で思わず涙がこみあげてきたぐらいなのだ。とにかくゼッタイにお勧めのAVGだと断言できる。

響 あきら(ゲーム評論家)
JICC出版局 別冊宝島63号「ミステリーの友」より

ジェリー・ランドルフはシカゴの新聞社、デイリー・カサブランカの女性記者である。ある日彼女の無二の親友メイ・エルガーが行方不明になり、それと時を同じくして、父エルガー博士の研究が軍部に狙われているという内容のメイの日記がジェリーのもとに届けられた。
早速、同僚のロイと共に彼女の家へ向かったジェリーは、研究室にある大きな機械の傍らで胸にナイフを受け、殺されている博士を発見する。調べると、研究所の裏庭には犯人のものと思われる足跡が残されていたが、不思議な事にその足跡は研究室に向かっているだけで出て行った形跡がない。犯人はどこへ消えたのか。また殺された博士の顔に浮かぶ穏やかな表情……。
あなたは犯人を追って過去へ飛ぶ。そこであなたを待ち受ける運命とは…?
AS TIME GOES BY—あなたが経験する愛と冒険の人生に乾杯!

シンキングラビットのディスクミステリー「カサブランカに愛を」は、SFを舞台にミステリーあり、ロマンスありのフルコースでお届けします。X68000のために、追加も含め全グラフィックを改良しました。3段階に変化する美しいグラフィックと、古き良きアメリカを彷彿とさせるBGMで、他のゲームでは味わえない珠玉のストーリーをお楽しみください。

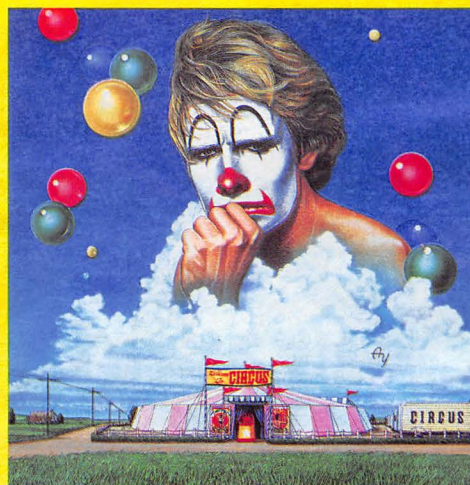
好評発売中 5インチ2HD ¥7,800

X68000

Meurtre d'un Clown.

道化師殺人事件

昭和60年4月に登場し、アドベンチャーゲームの方向を完全に決定づけた「道化師殺人事件」。発売後数年を経た現在でも、その伝説は衰えることなく、アドベンチャーゲームの金字塔として高く評価されています。
しかも、今までの機種では惜しくも表現できなかった部分が、X68000によって実現しました。
65000色カラーグラフィックモードを使ったフルスクロールによる臨場感と、練り上げられたシナリオが描き出す殺人劇。
アドベンチャーゲームの最高峰「道化師殺人事件」、ついにX68000に登場しました。「道化師殺人事件」を超えるのは、「道化師殺人事件」です。

好評発売中 5インチ2HD ¥7,800

X68000

ハードボイルドアドベンチャー

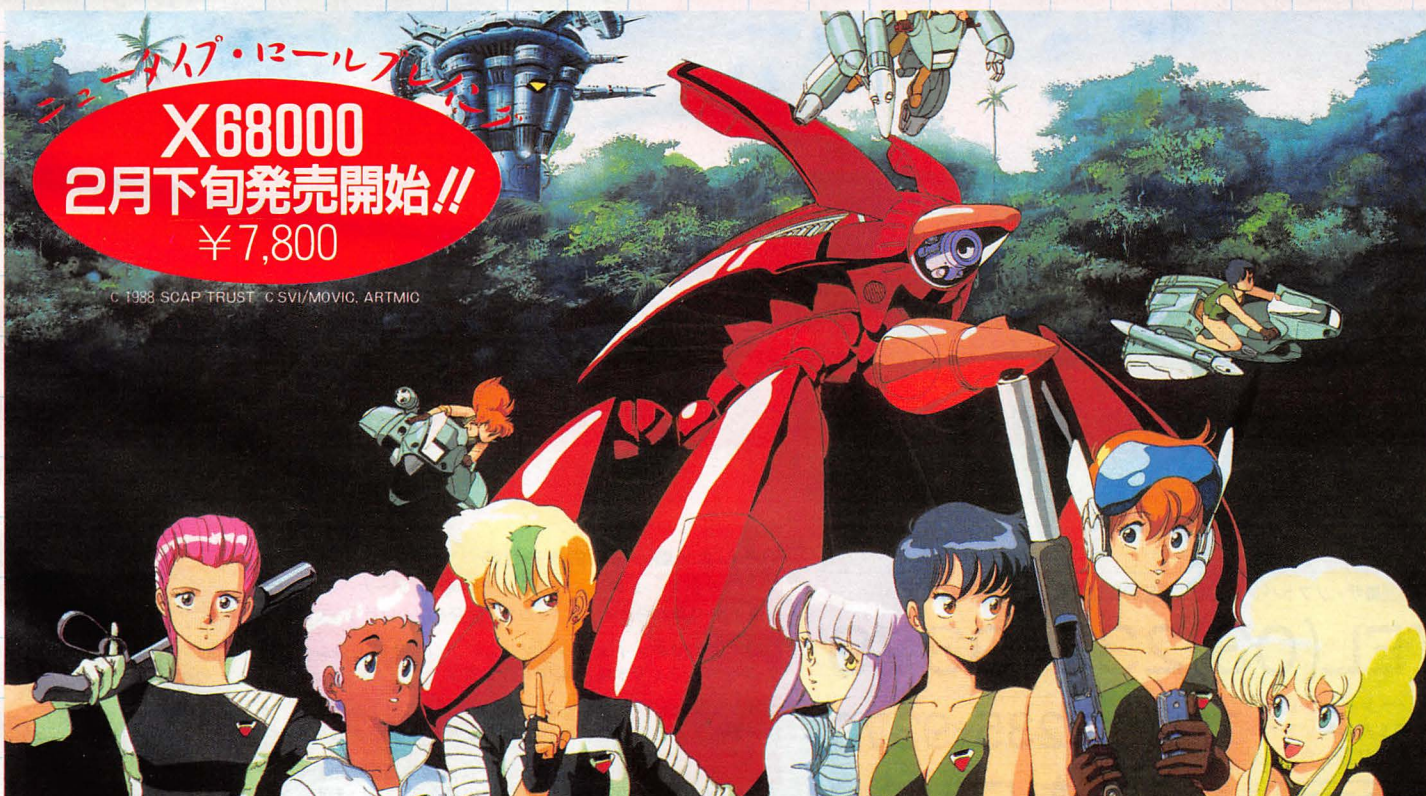
「ザ・マン・アイ・ラブ」来春発売!

GALL FORCE
ETERNAL STORY

ガルフォース『怒濤のカオス』

――未来を賭けた少女戦士の鼓動が、
いまキミを直撃する。――

話題沸騰中のゲーム『ガルフォース 怒濤のカオス』がついに発売される。新天地カオスをめぐる死闘ともいえる闘いに巻き込まれてしまった7人の少女戦士たち。無事にカオスに到着したのも束の間、敵の手からのカオス奪回の任務が彼女たちを待っていた。しかもパラノイド軍たちからの容赦のない攻撃が続く。新天地カオスに隠された大きな謎とは？そして彼女たちの運命は？すべての鍵をキミが握る。ロールプレイングゲーム『怒濤のカオス』。彼女たちの熱い鼓動がキミの心に響きだす。いま、未来を賭けた壮大なドラマの幕が開く。

ロールプレイングゲーム『怒濤のカオス』は、従来のゲームの枠を大きく超えたものになっています。奥が深いドラマチックなシナリオ。単なる情報収拾のためだけになりがちだった謎解きではなく、ゲーム世界とストーリー展開に大きく影響を与える謎がいたるところに配置されています。また、プレイヤーは7人の少女たち全員を操作することができ、キミの意志が彼女たちの運命を左右します。戦略的なおもしろさが味わえる戦闘シーン、キャラクター同士が、個性を活かした会話を交わすキャラクタートーキングシステムなど新しいRPGの領域へと踏み出したガルフォース『怒濤のカオス』。ついに、登場!!

X68000
2月下旬発売開始!!
￥7,800

スタッフ募集／グラフィック・プログラマー・シナリオライター・ゲームデザイナー(自宅作業可能)、及び作品持ち込み歓迎します。

ScapTrust 株式会社スキャップトラスト
〒150 東京都渋谷区神宮前5-42-1
ユーザーサポートテレホンサービス(03)486-8127

これで、″おタク族″の先輩をやめてください。

友だちに「おタク」と呼びかけるパソコン少年たちは、″おタク族″とも言われています。この言葉からイメージするのは、専門知識だけで頭でっかちになり、キーボートとCRTディスプレイしか視野にない子どもたち。私たち『日本ソフトバンク』は、コンピュータ好きな大学生に、そんな″おタク族″の先輩でいて欲しくありません。専門知識はもちろん、視野を広く業界に拡げて、情報通になっていただきたいのです。そのための情報誌「ザ・ソフトバンク」。最新の話題を満載しています。ぜひ、ご一読ください。

●″おタク族″の少年たちにも教えてあげたい「ザ・ソフトバンク」。残念ながら、未成年の方にはご遠慮いただいています。

無料だぜ！

the soft bank 3
ハイテク企業が無料で もの と 情報を プレゼントしちゃうパブリシティマガジン
コンピュータ関連企業
総特集

TETRIS

テトリス

好評発売中!!

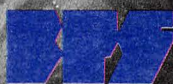
このくやしさは、快感だ。

シンプルなものほど奥が深い。それは、おもしろさの本質を見極め、磨きあげているからだ。ソ連で生まれたゲーム「テトリス」が、世界中で人気を博しているのも、本物のおもしろさを持っているからだ。ルールは簡単だが、その楽しさは奥が深い。ゲームオーバーになるたびに感じるくやしさが、やがては快感になるほど「テトリス」は魅力に満ちている。

好評発売中 ¥6,800

PC-8801シリーズ・X-1シリーズ・PC-9801(要256KBRAM)
PC88VAシリーズ・X68000・FM77AV・FMR-50
MSX2 専用 ♪ メインRAM64K、VRAM128Kを必要とします。

X1 シリーズの発売がおくれています。ご迷惑をかけて申しわけありません。

MSX はアスキーの商標です

ビー・ピー・エス 担当/吉田・南都
〒226 横浜市緑区鴨居3-1-3 鴨居ユニオンビル4F TEL045-931-0151

「テトリス」ゲームクイズ実施中▶詳しくはパソコンショップ店頭にあるチラシをご覧ください。

★★★★★★★★★ 1988年度 ★★★★★★★★★

GAME OF THE YEAR

ノミネート作品発表

お待たせしました，新春特別大サービス企画「1988GAME OF THE YEAR」のノミネート作品がついに発表です。1988年に登場した数多くのゲームのなかから優秀な作品ばかりをノミネートし，そのなかでもっとも優れた作品をみんなでよってたかって選び出し，でっかい賞をプレゼントしようという，Oh!X恒例の読者参加によるこの企画。アドベンチャーあり，シューティングあり，RPGありといった前年とは比べものにならないほどの大混戦のなかで，1988年度における真のGAME OF THE YEARに選ばれる作品を決定するのは，あなたの1票なのです。

1988 GAME OF THE YEAR

1988年度ゲームソフトの傾向と対策

1988年のゲーム界，これはすべてX68000を中心に展開されたといっても過言ではないでしょう。前年のスペースハリアーを超えろとばかりに，数多くのゲームが登場しました。そうして源平，ドラスピ，ドッジボールといったアーケード版ゲームが移植され，これまでにないまったく新しい世界を我々ゲームファンに教えてくれたのです。

しかし，もう一方では8ビット機も負けていません。イースII，ソーサリアンといったファルコム路線が圧倒的な強さを見せ，またレイドックやSUPER大戦略といった個性的なゲームも多くの支持を受けたのです。ゲームが不作だと嘆いていた前回と比べて，なんと楽しい状況になってきたことでしょう。遊びたいゲームが迷ってしまうほど揃っている，そういった贅沢なグチが飛び出すほどの明るい状況が見え始めた年，それが1988年のゲーム界だったのです。

まずは隣に並んだ，アンケートハガキ集計によるベスト50の結果を見てください。この集計は，この1年間に送られてきた愛読者のカードの「推薦する市販ソフト」欄を，前回の投票用紙を兼ねていた1月号を外して，2月号から12月号までの11カ月分から各1100通ずつ，計12100通を抽出して単純集計したものです。

見ていただければわかるように，なんと，日本ファルコムの強いこと。3位をこれだけ離してブッチギリの1，2フィニッシュです。そのほかではジャンルが見事なまでに分散しているのが特徴的。それだけ多岐にわたって数が揃っていたともいえそうです。なおこの集計にあたって，前年度発売のもの，またはゲームでないものは別表にまとめておきました。また，X1(turbo専用も含む)/X68000の両方に発売予定のものでも，11月末現在の時点で発売されているものということで機種名は記入してあります。

さて，比較的好調だった今回のGAME OF THE YEARですから，各賞の解説を見ていただければわかるように，その枠を大幅に拡大してお届けします。これらの賞の栄誉に輝くのはいったいどのゲームなのでしょうか。最後にある募集要項を確認のうえ，皆さんふるってご参加ください。発表は4月号，それまでお楽しみにー!!

前年度発売作品&その他ベスト20

1. スペースハリアー(X68000)……427
2. C compiler PRO-68K(X68000)……322
3. 上海(X1)……279
4. リバイバー(X1)……256
5. 三国志(X1)……191
6. MUSIC PRO-68K(X68000)……154
7. C-TRACE68(X68000)……127
8. 信長の野望・全国版(X1)……113
9. Z'sSTAFF PRO-68K(X68000)……93
10. Kamikaze/BUSINESS PRO-68K(X68000)……67
11. ジーザス(X1)……63
12. NEW Print Shop PRO-68K(X68000)……49
13. ウィザードリィ(X1/MZ-2500)……47
14. ダークストーム(MZ-1500)……46
15. 電脳倶楽部(X68000)……40
16. Sampling PRO-68K(X68000)……39
17. ドラゴンクエストIII(ファミコン)……30
18. R-TYPE(PCエンジン)……21
19. OS-9/X68000(X68000)……20
20. ルクソール(X1)……18

1988 GAME OF THE YEAR
アンケートハガキ集計によるベスト50(12100通集計)

1. イースII(X1)……1242
2. ソーサリアン(X1)……1224
3. 源平討魔伝(X68000)……858
4. スーパーレイドック(X1)……652
5. SUPER大戦略(X1)……531
6. ドラゴンスピリット(X68000)……495
7. マンハッタン・レクイエム(X68000)……315
8. Might & Magic(X1)……258
9. ハイドライド3(X1)……219
10. めぞん一刻・完結編(X1)……189
11. ぎゅわんぶらあ自己中心派2(X1)……187
12. ユーフォリー(X1)……175
13. スペースハリアー(X1)……174
14. 蒼き狼と白き牝鹿・ジンギスカン(X1/MZ-2500)……147
15. ワールドゴルフII(X1)……138
16. ウルティマIV(X1)……109
17. Mr.プロ野球(X1)……106
18. ラスト・ハルマゲドン(X1)……99
19. 沙羅曼蛇(X68000)……97
20. A列車で行こうII(X68000)……94
21. プロ野球FAN(X1)……88
22. スタークルーザー(X1)……81
23. Kiss of Murder(X68000)……75
24. アルカノイド2(X68000)……70
25. 熱血高校ドッジボール部(X68000)……63
26. 今夜も朝までPOWERFULまあじゃん(X1)……57
27. 琥珀色の遺言(X68000)……56
28. 抜忍伝説(X1)……55
29. サンダーフォースII(X68000)……52
30. たんば(X68000)……49
31. ウィザードリィ#2(MZ-2500)……46
32. ザ・リターン・オブ・イシター(X1/X68000)……43
33. 麻雀狂時代SPECIAL(X1/X68000)……40
34. ハウ・メニ・ロボット(X1/X68000)……39
35. 信長の野望・全国版(X68000)……37
36. 桃太郎伝説(X68000)……36
37. ドーム(X68000)……33
38. ファンタジーIII(X1)……30
39. ツインビー(X68000)……29
40. ソフトでハードな物語(X68000)……28
41. ゼリアード(X1)……25
42. Master of Monsters(X1)……24
43. 麻雀悟空(X1/X68000)……23
44. 第4のユニット2(X1)……20
45. T.D.F.(X68000)……19
46. 名監督II(X68000)……15
47. アークティック(X1/X68000)……14
48. ザ・コックピット(X68000)……9
49. アークス(X1)……8
50. ラプラスの魔(X1)……5

審査委員ノミネートベスト10

1. 熱血高校ドッジボール部(X68000)
2. A列車で行こうII(X68000)
3. スタークルーザー(X1)
4. マンハッタン・レクイエム(X68000)
5. SUPER大戦略(X1)
6. サンダーフォースII(X68000)
7. ユーフォリー(X1)
8. Mr.プロ野球(X1)
9. 桃太郎伝説(X68000)
10. たんば(X68000)

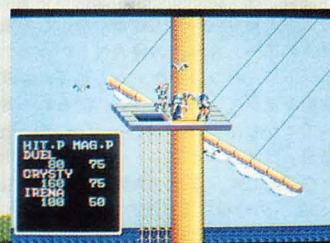
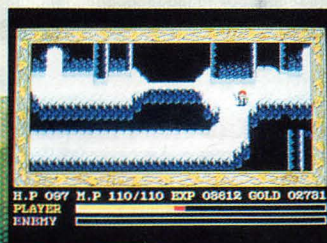
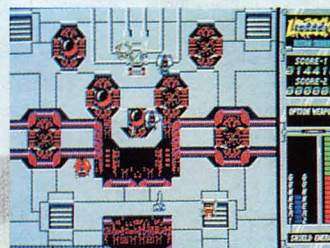
Oh!Xゲーム大賞

この賞は，昨年度までは「作品賞」としてあがめ奉られていた賞ですが，やはり年間を通じて最高の栄誉を得る賞には威厳が必要，ということで，こんなに立派な名前が付いてしまいました。しかし，毎年多様化するゲーム界において，いったいなにを基準にするか非常に難しく，それを暗示するかのように今年もさまざまなジャンルから20本のノミネートとなっています。

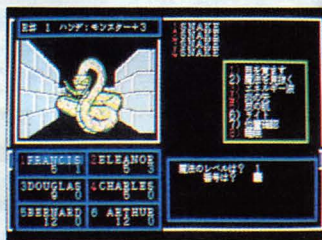
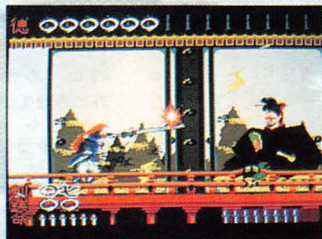
ここにズラッと並んだ作品を見てみると，まず最初に気づくのが，毎年この賞のノミネートの常連となりつつある日本ファルコムの強さ。もう，このソフトハウスはゲーム界の五木ひろしと化した感がありますが，そのパワーは今年も健在と言えそうです。

一方では，シミュレーションのジャンルでは現代戦略モノと時代戦略モノの両巨頭が仲よく「SUPER大戦略」と「ジンギスカン」でノミネート。アクションRPG全盛の時代にどこまでがんばるか期待したいところ。そしてあとはやはりX68000パワーの凄さにご注目。「源平討魔伝」，「マンハッタン・レクイエム」，「A列車で行こうII」，「ドラゴンスピリット」と堂々4本もノミネート。発売時期があれほど遅れた「ドラゴンスピリット」が，ここまで健闘したのには驚かされます。

そして最後に，「Mr.プロ野球」と「ユーフォリー」といった地味ながら楽しませてくれた，実力派ゲームがどこまで票を伸ばすか，これも大いに注目したいところです。

蒼き狼と白き牝鹿・ジンギスカン（X1/MZ-2500）
イースII（X1）
A列車で行こうII（X68000）
ぎゅわんぶらあ自己中心派2（X1）
源平討魔伝（X68000）
スタークルーザー（X1）
スーパーレイドック（X1）
SUPER大戦略（X1）
スペースハリアー（X1）
ソーサリアン（X1）
ドラゴンスピリット（X68000）
ハイドライド3（X1）
プロ野球FAN（X1）
Might & Magic（X1）
マンハッタン・レクイエム（X68000）
Mr.プロ野球（X1）
めぞん一刻・完結編（X1）
ユーフォリー（X1）
ラスト・ハルマゲドン（X1）
ワールドゴルフII（X1）

ゲームデザイン賞

この賞は，グラフィック・音楽・操作性など，ゲームを構成する各要素のバランスのとれた作品に贈られる賞です。ここでもやはりと言おうか，日本ファルコムの2本がノミネート。それと独創的なコンセプトでデザインされている「A列車で行こうII」と，アーケード版から移植の人気ゲーム「源平討魔伝」といったX68000グループも，ハードとの相性もよく完成度の高さではかなりのもの。また，アメリカらしい完成度を誇る「Might & Magic」の個性も捨てがたいところです。

イースII（X1）
A列車で行こうII（X68000）
源平討魔伝（X68000）
ソーサリアン（X1）
Might & Magic（X1）

オリジナルシナリオ賞

いまやAVGやRPGでは，シナリオのレベルの高さはもはや必需品。ゲームをよりいっそう盛り上げるというより，もうシナリオ自体が推理小説や冒険小説として通用するところまできているようです。なかでも，リバーヒルソフトのこの1年の活躍には目を見張るものがあります。

それと「イースII」のあのテンポのいい展開も，ずいぶんと多くのユーザーを楽しませてくれました。「ラスト・ハルマゲドン」の壮大なストーリーは，人類滅亡の謎と魔族という新しい関係を見事なまでに表現しています。このノミネート作品を見ていると，1988年は本当の意味でシナリオの地位を，ゲームの世界でも確立してくれた年であったといえそうです。

イースII（X1）
琥珀色の遺言（X68000）
マンハッタン・レクイエム（X68000）
ラスト・ハルマゲドン（X1）

テーマ音楽賞

音楽賞といっても，某テレビ局の金の鳩賞なんかとは違ってゲーム音楽の賞ですので，音楽性云々よりもストーリーの展開に曲が非常にマッチしているとか，シナリオの雰囲気をより盛り上げてくれるなどの要素をもとに選考されています。

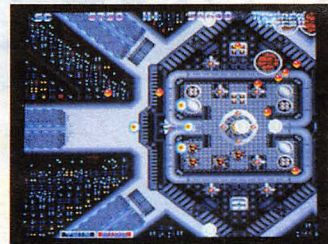
それはともかく，今回は音楽に関しては凄い年だったといえそうです。なかでも「琥珀色の遺言」のオープニングなんかは，もうゲームが始まる以前にプレイヤーをゾクゾクさせてくれる魅力を持ったものでしたし，「イースII」や「ソーサリアン」のパソコンを使ったゲーム音楽としての完成度の高さには，ただただ感心させられるばかりです。このように一段とレベルの高い激戦区ともいえそうです。

アークス(X1)
イースII(X1)
琥珀色の遺言(X68000)
ソーサリアン(X1)

グラフィック賞

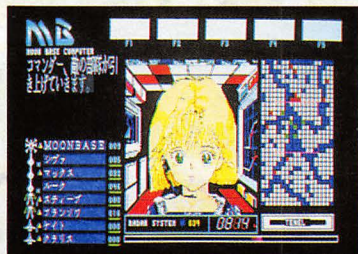
前回までは比較的，「美しいグラフィック」というのがこの賞の重要なポイントとなっていましたが，やはりX68000が登場してからは，それに加えて，「いかにきれいに見せてくれたか」，といった絵のセンスというのもポイントです。

ですから今回は，「たんば」や「サンダーフォースII」といった，絵のセンスが買いのゲームと，オープニングだけでも多くの人間をあっと言わせた「マンハッタン・レクイエム」などが中心となってノミネートされています。この賞のノミネート作品を見てみると，やはりX68000が与える影響は，今後ますます大きくなってきそうです。

イースII(X1)
サンダーフォースII(X68000)
たんば(X68000)
マンハッタン・レクイエム(X68000)

特別演出部門賞

最近はゲーム全体のレベルも上がり，もはや思いつきでプログラムを組んで完成させるといったことは通用しなくなってしまったようです。そうして制作者はゲームをプレイヤーにいかにアピールするかといった，技術が要求される時代へと移り変わってきました。この賞は，見せ方の工夫，作り方の優秀性といったことをポイントに，パソコンゲームとして，より優れた演出を見せてくれたゲームに贈られる賞です。

ですから，「スタークルーザー」や「琥珀色の遺言」は画面レイアウトやムードの演出に凝った作品として，また「A列車で行こうII」はX68000への移植によりその世界をますます完成させたゲームとして，さらに「熱血高校ドッジボール部」や「たんば」は，ドッジボールや霊界スゴロクといった題材をうまくパソコン流に料理した，といった観点からノミネートされました。

A列車で行こうII(X68000)
琥珀色の遺言(X68000)
スタークルーザー(X1)
たんば(X68000)
熱血高校ドッジボール部(X68000)

主演キャラクター/助演キャラクター賞

今回のノミネートを見て，「主演キャラクター賞のなかになんでA列車がニッコリ笑っているんだよー」と驚かれている貴兄もいらっしゃることと思います。でも，よーく考えてみてください。ゲームのなかで一番重要で，なおかつ愛着を持って接することのできる要素を持ったもの，というのがこの賞の選考基準なのです。大自然のなかをプレイヤーの手足となってせっせとレールを敷いて走るA列車って，やっぱりカワイイ個性を持ったキャラクターのはず。人間じゃなきゃいけないわけもなし，というわけで今回，初めて列車がノミネートとあいなったわけなのです。

このほかにも主演・助演の両方に登場したドッジボール部のメンバーや，オープニングではとてもカッコイイのに，いざゲームが始まるとバニーちゃんみたいなかわいいスタイルで暴れ回る「ゼリアード」のデューク・ガーランド，そしてプリプリとした質感がとてもいいと好評のX68000版イシターのスライムと，今回はずいぶんと変化に富んだ豪華メンバーの登場となっています。

なお，パソコンゲームの主演・助演キャラということを念頭に置いての選考となっていますので，「めぞん一刻」の出演メンバーはあえて選外とさせていただきました。

主演キャラクター賞

アドル(イースII・X1)
A列車(A列車で行こうII・X68000)
カイ(ザ・リターン・オブ・イシター・X68000)
景清(源平討魔伝・X68000)
くにお(熱血高校ドッジボール部・X68000)
デューク・ガーランド(ゼリアード・X1)
桃太郎(桃太郎伝説・X68000)

助演キャラクター賞

安駄婆(源平討魔伝・X68000)
スライム(ザ・リターン・オブ・イシター・X68000)
ソニー君(ぎゅわんぶらあ自己中心派2・X1)
天使(熱血高校ドッジボール部・X68000)
リリア(イースII・X1)
レイチェル(ストーム・X1)

Oh!MZ賞

やはり毎年寂しくなってくるこの部門。これも時代の流れといってしまえばそれまでなのですが，光栄の最新作「蒼き狼と白き牝鹿・ジンギスカン」の登場に，少しは救われたような気がします。またあの名作，「ウィザードリィ＃2」の登場でMZ-2500には質の高いゲームが揃ったともいえそうです。最後はやはりこの人，古籏氏のMZ-700版スペハリも自作プログラムながら，ノミネートのなかに名前を連ねてもらうことにしましょう。これはやはりOh! Xならではの部門なのですから。

蒼き狼と白き牝鹿・ジンギスカン
ウィザードリィ＃2
MZ-700版スペースハリアー

Oh!68賞

前回，スペハリの登場に真っ青になってしまい，急遽設定してしまったという，X68000の実力を思い知らされたことはまだ記憶に新しいように思います。でも，今回はそういったにわか仕込みのブームに流されることもなく，着実にその数を増やし，いまや「X68000にはオリジナルゲームしか作らない」とまでソフトハウスからいわれるようになりました。

そんななかで，今回はアーケード版ながらもパソコンになかなかマッチした仕上がりの「源平討魔伝」。完全な移植版とまでいわれた「ドラゴンスピリット」。そしてオリジナルシューティングの最高峰「サンダーフォースⅡ」の3本がノミネートです。スペースハリアー以来，電波の作品にはますます磨きがかかってきたようです。いずれにしてもこの部門，各ソフトハウスの熱意が伝わってくる力作ばかりです。

源平討魔伝
サンダーフォースⅡ
ドラゴンスピリット

Oh!X賞

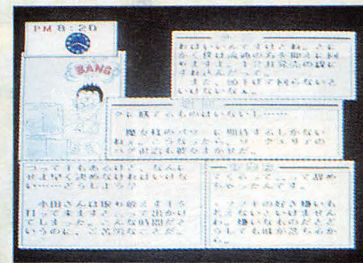
シューティングゲームが出てこない，と騒いでいるところに突如として登場し，若い世代のX1ユーザーの圧倒的支持を得た「スーパーレイドック」。そして，斬新なアイデアで新しいシステムを構築してくれそうな予感を与えてくれた「スタークルーザー」。「SUPER大戦略」は細かいところまでしっかりと作られた大作ということでノミネートです。とにかく，どの作品もX1のゲームとしては最高級。いずれにしても，どれが賞を取ってもおかしくないゲームばかりなのです。

SUPER大戦略
スーパーレイドック
スタークルーザー

ファンキーアイデア賞

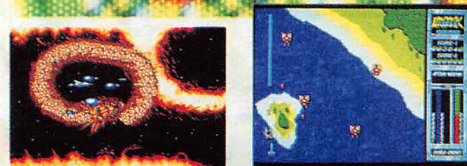
今回からスタートするこの部門。簡単に説明すると，プレイする側からは想像もつかないようなアイデアをゲームに持ち込んで，思いっきり我々を楽しませてくれたゲームに贈られる賞です。

とにかく，「たんば」のあのグラフィックと相原コージのイラストには強力なインパクトがありました。また，アドベンチャー，シミュレーション，そしてタイムトライアルと3つの要素の幕の内弁当，「ストーム」は，ほかでは決してマネのできない個性を持っています。ゲームとしてのトータルな意味での完成度の高さも必要ですが，こうして心がなごむような単純明快な楽しさも，これからのゲームには期待したいところです。

紫醜罹(X1)
ストーム(X1)
ソフトでハードな物語(X68000)
たんば(X68000)

シューティングゲーム賞

前回は該当作品の数が揃わなかったために，一時お休みをいただいていたこの部門ですが，今回は年明け早々X1に登場して話題をまいたあの「スーパーレイドック」を始めとして，X68000がいよいよその底力を見せ始めた証として，「ドラゴンスピリット」や「沙羅曼蛇」，「ツインビー」，そして最新作の「サンダーフォースⅡ」までと，いきなり豪華なラインアップとなってしまいました。

ここはやはりアーケード版からの移植モノより，パソコンオリジナルの健闘を期待したいところです。ただ，最初にご紹介した愛読者カードの単純集計によるベスト50を見ていると，未だにX68000版スペハリの与えたインパクトの大きさはその数字となって表れていることがよくわかります。しかし，残念ながら前年度の作品ということで選外となっています。その代わり，弟ともいえるX1版が堂々のノミネートです。

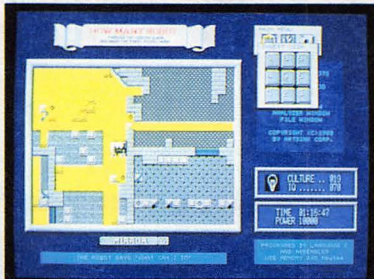
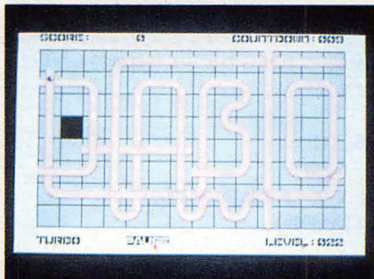
沙羅曼蛇(X68000)
サンダーフォースⅡ(X68000)

スーパーレイドック(X1)
スペースハリアー(X1)
ツインビー(X68000)
ドラゴンスピリット(X68000)

パズルゲーム賞

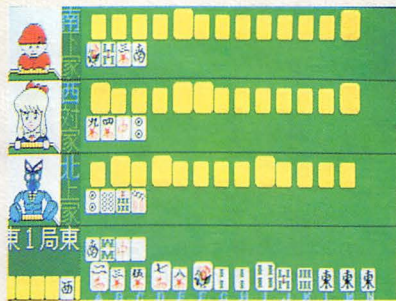
前回まではインテリジェント賞と呼ばれていたこの部門ですが，今回はそれぞれのジャンルが豊富に揃ってきたために，パズルとテーブルゲームと２つのジャンルに分かれての登場です。

とにかく，ノミネート作品をご覧になってください。「アークティック」，「ハウ・メニ・ロボット」とアートディンクの個性が光っています。やはり，このようなジャンルには，各ソフトハウスの持つ，独特の個性が現れるといえるでしょう。

また，後発ながらも，リアルタイムでブロック崩しの逆をいく「TETRIS」や，転がるボールに手に汗握る「DIABLO」も，基本に戻った楽しさを与えてくれる名作でした。

アークティック(X1/X68000)
DIABLO(X1)
TETRIS(X1/X68000)
ハウ・メニ・ロボット(X1/X68000)

テーブルゲーム賞

パソコンの思考ゲームのなかで，やはり昔から不滅の領域を持って存在するのが，このトランプや麻雀，花札といった卓上ゲームの数々。そのゲームのなかから卓越したゲームを選抜してお届けするのがこのテーブルゲーム賞です。

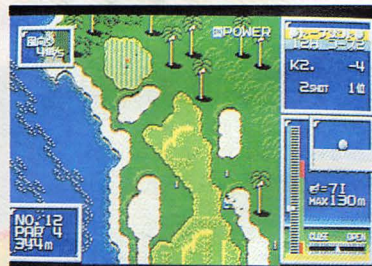
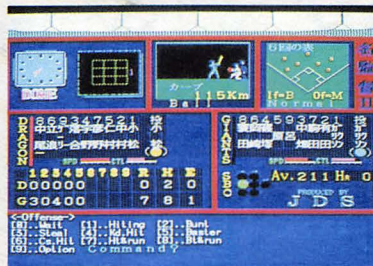
今回のノミネート作品の顔ぶれを見ていると，麻雀ゲームのなかにもこれまでと違った方向性が見え始めたのが特徴のようです。たとえば「ぎゅわんぶらあ自己中心派２」の個性の強いキャラクターたち。または，ぽこ麻雀を取り入れた「今夜も朝までPOWERFULまあじゃん」などのように，単なる麻雀ゲームの要素に，さらにエンタテイメント性を加えたものが主流となりつつあるようです。この部門も今後どこまで新しい展開を見せるか楽しみなところです。

ぎゅわんぶらあ自己中心派２(X1)
今夜も朝までPOWERFULまあじゃん(X1)
ソリティアロイヤル(X1)
花札放浪記(X68000)
麻雀悟空(X1/X68000)
まあじゃん狂時代SPECIAL(X1/X68000)

スポーツ大賞

この部門は，某テレビ局の番組とは一切関係ございません。別に編集者が野球大会に参加するわけでもなく，ただなんとなくゴロがいいからと並べてみたらそうなってしまったわけなのです。

前置きが長くなってしまいましたが，これも今回は豊富なジャンルのゲームがたくさんあったからこその設置です。しかし，ここではメジャーなジャンルのスポーツゲームを中心に攻めてみたいとの希望もあり，あえてアクション指向の強いドッジボールには外れてもらうことにしました。

そんななかで今回特徴的だったのが「Mr.プロ野球」，「名監督II」といった監督業を中心としたスポーツシミュレーションが登場してきたこと。これはファミコンに対抗するためのスポーツゲームとして，新しい展開を見せてくれようとする意気込みが感じられるようで，ますます発展してほしい気がします。特に「Mr.プロ野球」は，タケルソフトでこの健闘には目を見張るものがあります。

プロ野球FAN(X1)
Mr.プロ野球(X1)
名監督II(X68000)
ワールドゴルフII(X1)

移植海外ゲーム賞

日本のパソコンゲームも，ハードの進歩とともにレベルも毎年向上しています。しかし，海外のゲームの持つ独創的な発想も捨てがたいところ。というわけで今回もこの部門は健在なわけですが，なんといっても注目株はソビエトからのお客さん「TETRIS」の登場です。

これまでこのような海外からの移植ものといえば，そのほとんどがアメリカ産RPGだったわけですが，意外なことにソビエト産のパズルゲームがノミネートされることとなりました。これもソウルオリンピックの影響なのでしょうか。そのほかには，やはりスタークラフトの移植RPGが例年どおりの強さを発揮しているようですが，いつものX1/X1turboの両方のサポートに加え，嬉しいことに最近はX68000版も発売されるようになりました。

TETRIS(X68000)
ファンタジーIII(X1/X68000)
Might & Magic(X1/X68000)

最優秀パフォーマンス賞

今回，突如として登場のこの部門。これはゲーム以外の部分，特に店頭デモ用ソフトなどで盛り上げてくれたゲームに贈られる賞です。こういったデモ版というのは，比較的皆さんには馴染みがないかもしれませんが，最近，店頭デモや付録に付いているデモ版といった，デモ用ソフトのレベルが，非常に高くなってきているのです。

これまでデモ版というと，オープニングやゲーム中の画面を一部切り取っただけというものが主流だったのですが，最近はこのデモ専用のソフトを新しく作成するソフトハウスが多くなり，その内容もグラフィック，音楽，そして演出という要素がうまくマッチして，わずか数分間のものでありながら，もうそれだけで1本の環境用ソフトとなり得るようなものが登場し始めています。ですから，そのようにわずか数分間に楽しませてくれるようなデモ版ソフトのセンスを，これから大いに評価していこうと設置されたものです。

これには，X68000の登場が大きく関係しているといえそうです。しかし，それだけではX68000の独壇上になってしまうので，その枠を多少広げて，X1版「アークス」に付いていた「YASHA」のデモソフトと，同じくX1版の「ラスト・ハルマゲドン」の12分にもわたるオープニングもノミネートされています。このように，パソコンゲームのデモ版が，映画の予告編を超えてしまうかもしれないような時代がもうすぐそこまで来ているのです。

A列車で行こうⅡ(X68000版店頭用デモ)
琥珀色の遺言(X68000版店頭用デモ)
殺人倶楽部DX(X68000版店頭用デモ)
YASHA(X1版アークスの付録)
ラスト・ハルマゲドン(X1版オープニング)

底抜け脱線ゲーム賞

いまさらこの部門に関しての細かい説明は不要でしょう。簡単にいえば「せっかく楽しみにして買ってきたのに，なんだこの○○○は」という，あの例のパターンです。詳しいことをお知りになりたい方は，1988年4月号の23ページをお読みください。

前回は比較的，まだ活気が足りなかったように思います。今回はもう少し気合いを入れていきましょう。でも頭にくるあまり，思いきり暗くなってしまうと困るので，パァーッと明るく笑えるものを重点的に掲載したいと思っています。ですから，推薦理由というのをわかりやすくお書きください。今回は，スペースも前より大幅拡大の予定です。

ノミネート作品なし

その他自由応募部門賞

この賞はその名のとおり，まったく自由に皆さんが思いつくままに設定していただいて結構な賞です。前回は，当然ながらゲームかソフトハウスだけのものだろうと思っていたら，「横綱双羽黒」からイラストの「田村，高橋コンビ」，そして「地球防衛軍の広告のお姉さん」までと，ほとんど収拾のつかない無法地帯と化してしまいました。今年もこの調子でガンガンいきましょう。こちらもスペースを拡大してお届けする予定です。

◆5月に発売されたソーサリアンは，X1turbo専用だったにもかかわらず，かなりヒットしたという。確かに，最初のシステムに付属のシナリオをPart.1だとすると，この12月に発売のピラミッド・ソーサリアンのPart.4まで，実に4本のシナリオがこの半年あまりの間に発売されたことになる。よく「～2」とか「～3」までが2年ほどの間に発売されたという話は聞くが，半年で4本というのは異例のことである。

このソーサリアンは，チップ（16×8ドットの絵を構成する最小単位）で描かれたとは思えないような美しいグラフィックと，すぐにCDが発売されてしまうほどの素晴らしい音楽，そしてシナリオと，どれをとっても最高級のできである。

よく，「最近のゲームは同じようなものばかりでつまらん」という人がいるが，このソーサリアンも胸を張っていばれるほどの“あったらしいっ！”アイデアが盛り込まれているというわけではない。しかし，追加シナリオが発売されるたびに私を「ペンタウァ」の町に呼び戻し，独特の世界を堪能させてくれているのも事実なのだ。このソーサリアンが，これからいったいどのような新しい展開を見せてくれるか期待してみたい。（西川善司）

◆カタギの人々はとっくに寝静まり，凍てつく隙間風が身体によくない季節。なに気なく立ち上げたゲームが妙に面白くて，つい明日のことを忘れる。そんなゲームが，いくつかあったですね，この1年は。ただ，そのなかで1本となると，うーん，さて？

X1系とX68000の2つにばっさりと分かれてしまったOh！Xゲーム界。RPGの逆襲，ラスト・ハルマゲドン。スポーツの復活，ワールドゴルフⅡ，Mr.プロ野球，ドッジボール部。アクションは不滅，スーパーレイドック，源平討魔伝，サンダーフォースⅡ。あの感動を再び，A列車ⅡやSUPER大戦略。ADVもいるぞの琥珀色の遺言。イースとソーサリアンばかりがなぜ人気なのかわかんないくらい，1988年は多種多様なゲームが暗躍した。

私のお気に入りはワールドゴルフⅡであり，（獲得賞金総額は10億円を越えた），A列車Ⅱであり，サンダーフォースⅡであった。あれ？全部Ⅱものだぞ。ま，いいか。

今年は，確かにゲームはたくさんあった。しかし，新しいゲームが登場した年ではなく，いままでの経験を元にして完成度の高いゲームが出現した年だといえそうだ。新作のTETRISやら，たんばやら，Mr.プロ野球など新しいものも，あるにはあったが，熟練ジャンルの牙城は崩せなかったということか。ソビエト製のTETRISがどこまで票を伸ばすかが楽しみ。この年末はSUPER大戦略68Kでもやって遊んでよっと（この姿勢がいけないな，最近どーも）。

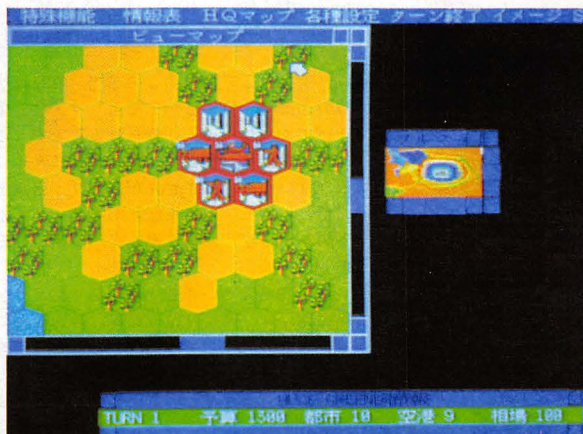
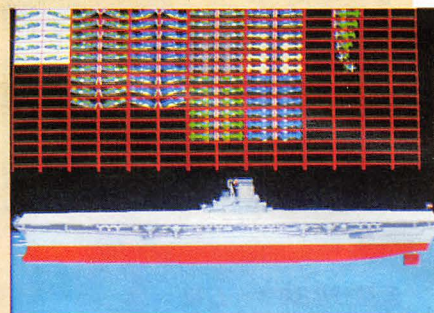
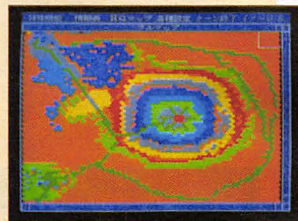
（荻窪　圭）

応募要項

1988 GAME OF THE YEARに応募されたい方は，今月号の愛読者カードの記入欄，または官製ハガキに各賞の名前とソフト名，そして推薦理由をお書きになって，Oh！X編集室までお送りください。

お送りいただいたハガキのなかから，メッセージを採用させていただいた方に抽選で，写真の「ゲームミュージックCD」と，バンダイの「スーパーデフォルメ・ガンダムプラモデル」，そしてUPU発行の『電視遊戯大全・テレビゲーム』を各5名の方に，また100名の方にはOh！X4月号の表紙をあしらった「Oh！Xノート」をプレゼントします。締め切りは2月15日（当日消印有効）までです。皆さん，ふるってご応募ください。

SOFTWARE INFORMATION

DRAGON
リヴォルティー2
ソーサリアン追加シナリオVol.3
エグザイル2
株価分析ソフトCK-2
FMシンセサイザー・ボード/D.M.S.R.

まずは上の2つがSUPER大戦略68K。そして左下が今回はちょっと98版をお借りしての3Dゴルフ。そして最後がWINGS OF FURYの各開発途中の姿なのです

話題のソフトウェア

今月号の発売はまだ12月なのですが，Oh! Xはいち早く新年を迎えてしまいました。1988年度GAME OF THE YEARのノミネート作品も発表されたことですし，また1年よろしくお付き合いください。

というわけで，今月の「話題のソフトウェア」は，これまでとはちょっと趣向を変えて，まずは今月(1988年12月)，または次の春ごろまでに発売される予定の，新作ソフトの経過報告からすることにしましょう。上の写真を見てください。これが新年に登場する大作3本です。

まずはシステムソフトのSUPER大戦略68K，そしてお次はT&Eの3Dゴルフ，そして最後がブローダーバンドジャパンのWINGS OF FURYです。大戦略のほうはまだマップデータの作成中みたいだし，3Dゴルフは春ごろになるだろうといってるところへ，こっそりとうちのBeep取材班が潜入して入手した超極秘(?)写真なのです。おかげで，この3Dゴルフは残念なことにX68000版ではなく，98版しか入手できなかったそーです。Beepの取材班によると，この3Dゴルフ，T&E社内でも評判になっているほどマイペースで開発が進んでいるらしく，X68000版に至っては，発売日がぜんぜん決まっていないというのがどうやら真相らしいのです。まっ，それだけ完成度が高いゴルフゲームが出来上がると思

読者が選ぶゲームベスト10

今月の“1988 GAME OF THE YEAR”のノミネート発表，いかがだったでしょうか。あれもこれも思い出深い作品ばかりで，選考のほうもずいぶんと難航したようです。しかし，よく考えてみると，この「読者が選ぶゲームベスト10」も，もう次回のGAME OF THE YEARに向けての参考資料となるわけで，1年の経つのは本当に早いものです。

では，今月のベスト10を見てみましょう。なんとドラゴンスピリットが，イースII，ソーサリアンを抑えて，いきなりトップに躍り出ました。そのほかにも琥珀色の遺言やX68000版信長の野望といった，初登場で上位に入り込んでくるゲームが今月は目立っています。さて，今年はいったいどのような激戦が繰り広げられるのでしょうか。X68000の動きを中心に，大いに期待していようではありませんか。

1. ドラゴンスピリット
2. ソーサリアン
3. イースII
4. 琥珀色の遺言
5. ラスト・ハルマゲドン
6. SUPER大戦略
7. スタークルーザー
8. 信長の野望・全国版(X68000)
9. 沙羅曼蛇
10. サンダーフォースII

って，のんびり待ってみることにしましょう。

そして最後が，ブローダーバンドのWINGS OF FURY。どうです，この航空母艦のグラフィック。まるでプラモデルのパッケージでも見ているような美しい出来でしょ。こちらは，どうもPrint Shopの制作を手掛けて以来，社内にX68000ファンのプログラマさんが誕生してしまったようで，ずいぶんと気合いが入っているみたい。おまけに原作とストーリーをそっくり入れ替えて，ゼロ戦がアメリカ軍を撃退するという内容になっての移植版空中戦シミュレーションです。

さて，そのほかのゲームといえば，ソーサリアンの追加シナリオVol. 3の発売決定とともにますます期待がかかる**イースIII**。そしてシャープから発売予定のX68000版**パックマニア**と**サンダーブレード**。この両方とも詳しいことをお届けするまでにはもう少し時間がかかりそう。このほかにも現在シャープ・SPSラインでいくつかプロジェクトが進行中らしいので，これはまた次回にでもレポートすることにしましょう。

新作ソフト情報

☆……11月30日現在発売中　★……近日発売予定

★DRAGON

MZ-2500用パズルゲームの登場だ。このゲーム，画面上にズラリと並んだ麻雀牌のなかから，まずは隣り合った同じ牌を裏返す。そうして今度はすでに裏返された牌をはさんだ牌のなかから同じものを見つけて裏返していく。これを続けて全部の牌を裏返してしまえば，面クリアでどこかで見たような面クリドラゴンが待っているという寸法。うまく順序さえ考えれば簡単にいきそうだが，そうは問屋が卸さない。ゲームが単純なだけに，ハマッてしまう危険性が大きそう。

MZ-2500用　3.5″2DD版　6,800円
ログ　☎03(837)2595

★リヴォルティー2

X1には久しぶりのシューティングゲーム，リヴォルティー2が登場だ。X1シリーズのアクションゲームでは珍しい2ドット縦スクロールを実現。パワーアップオプションは10種類以上，BGMはすべてFM音源対応で20曲以上という超豪華版。ただし，このゲームはジョイスティック専用となっている。ゲームのストーリーは，アンドロメダ銀河のあるG型太陽を主星とするスヌクラ星系の第5惑星デヴィタに突如起こった，ベルネリ軍のクーデター。そうして平和に暮らしていた人々は，弾圧に苦しめられる暗黒の時代を迎えようとしていた。しかしそんななかにひと筋の光明が。それは，ベルネリ政権に立ち向かおうとする勇者たちの姿であった……。

X1/X1turbo用　5″2D版4枚組　予価7,800円
(2ドライブ専用，要ジョイスティック)
風雅システム　☎0764(29)6791

★ソーサリアン追加シナリオVol. 3

ついに，追加シナリオも第3巻の発売を迎えることとなった。今度の舞台は南海の孤島。日本での

DRAGON

戦いに勝利してペンタウァに帰国したソーサリアンは，休む間もなく王からギルバレス島での大魔王ギルバレスの復活を知らされ，新たな戦いの旅に出る。こうして追加シナリオVol.3「ピラミッドソーサリアン」は，謎のピラミッドに始まり，ラ・フォーヌの森，洞窟，巨大神殿，そして，戦国ソーサリアンの徳川家康の章で暗黒神「邪鬼」を操っていたギルバレスの迷宮の奥深くへと進んで行く。

X1turbo用　5″2D版2枚組　3,800円
(要ソーサリアンゲームディスク)
日本ファルコム　☎0425(27)0555

★エグザイル2

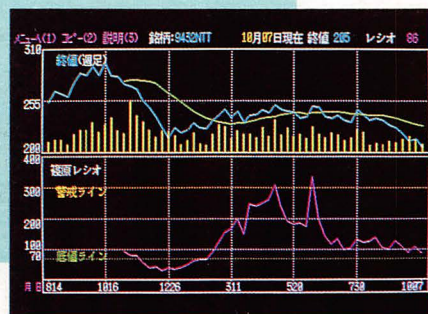
テレネットの中東を舞台としたアクションRPGゲーム「エグザイル」の完結編，エグザイル2の発売が決定した。このエグザイル，まだ11月の時点では最初のゲームのX1版は発売されていないが，他機種の仕上がりを見てみるとなかなかのもの。しかし，これが2ともなると操作性，グラフィックデータ量，ドラッグの効果，マギックすべてが前作を上回っているそうで，完全に前作を超えることが期待できそうだ。ストーリーは，世界を掌握した何者かが，いよいよ最後の構築物の建造にかかろうとし始めた。こうして，前作でアサシンの首領となった主人公サドラーを待ち受ける運命の扉が再び開かれる。果たして巨大な歴史の命運を握るのは誰なのか……。

X1turbo用　5″2D版4枚組　8,800円
日本テレネット　☎03(268)1268

☆株価分析ソフトCK-2

テラダ商電からすでに発売中の廉価版株価分析ソフト，CK-1を大幅にグレードアップしたCK-2が発売された。このCK-2は，ローソク足，出来高の表示，サイコロジカルラインの表示といったCK-1の持つ諸機能はもちろんのこと，さらにカイリ率，篠原レシオ，ボリュームレシオといった高度な分析も可能となった。このなかにあるカイリ率は株価移動平均線からどの程度かい離しているかを示す率，篠原レシオはテクニカルアナリストの篠原正治氏の開発した分析法で高値，安値，終値の動きから計算し特に仕手株の分析に向いているといわれるレシオ，ボリュームレシオは株価上昇日の出来高合計を株価下落日の出来高合計で割ったものである。このソフトではそれぞれの分析画面において分析法の簡単な説明が表示されるのでマニュアルがなくとも分析結果を把握することが可能。また上記の分析すべてを1画面に表示する総合分析の機能も持っているので分析がよりわかりやすくなる。

CK-2には3銘柄のデータが入力済みのデータディスクが付属しているのでそれを見ながら即使用が可能。そのデータフォーマットはCK-1と同一な

リヴォルティー2

株価分析ソフトCK-2

ので，すでにCK-1を使用しているユーザーはCK-2への乗り換えができる。

データは週足なので週末に一度だけ入力すればよく手間が省けるが，テラダ商電ではCK-2を使用したデータ提供サービスも有料にて行ってくれるとのこと。

X1turbo用　5″2D版2枚組　30,000円
(2ドライブ専用)
テラダ商電　☎0542(78)8662

☆FMシンセサイザー・ボード/D.M.S.R.

MZ-2500に新しく「FMシンセサイザー・ボード」と，ミュージックソフト「D.M.S.R.」が発売された。FMシンセサイザー・ボードは，FM音源部の基本スペックは，2オペレータによる6音ポリフォニック＋リズム音5音で，5パラメータのデジタルエンベロープとなっており，ヤマハ製のミュージックキーボードYK-01，もしくはYK-20に接続でき，手軽に演奏が楽しめる。

またこのボードには基本ソフトも付属しており，64音色がプリセットされている音色エディタと16パターンのリズムがプリセットされたリズムエディタによって，音色とリズムの編集が可能。1小節を最大32ビートで表現できるこのリズムエディタは，パターン作成用マトリクスによって演奏リズムパターンを簡単に作ることができるようになっている。

一方のD.M.S.R.は，FMシンセサイザー・ボード用のデジタル・マルチシーケンスレコーダで，FM音源チップをフルに使った9音ポリフォニックとMZ-2500内蔵のFM音源を使用したリズム音3声による自動演奏を可能にした作曲用ソフトで，メロディはオプションのミュージックキーボードからのリアルタイム録音，リズムはパターンエディタによって作成し，リズムエディタで最大64パターンまで管理できる。なお，このD.M.S.R.の使用にあたっては，増設RAM/VRAMが必要。

FMシンセサイザー・ボード　24,800円
D.M.S.R.　3.5″2DD版　9,800円
ニッコーシ　☎03(270)8851

GAME REVIEW

今月は，固定ファンの多いAVG「第４のユニット」のX68000版と，さらに個性的な「極道陣取り」，そうしてコミカルホラーAVG「白夜物語」です。どうやら今回は，笑いが絶えない３本セットになってしまったようです。

第4のユニット

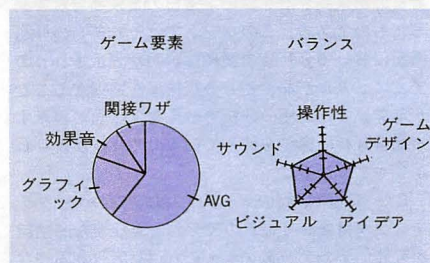
記憶を失った少女の過去に隠された，巨大な陰謀の影。ユニークな戦闘シーンに思わず笑ってしまうAVGなのです。

▶まず，８ビット版のこのゲームとの比較ですけど，かなりよくなりました，うん。さすが，もともとグラフィックツールを作っていた（G-EDITとかですね）ソフトハウスの作品だけあって，スピードの点でも綺麗さという点でも，X68000のグラフィックの機能をよく使いこなしていると思います。ただ，もう少し派手な色使いでもよかったかなーというのと，少しぐらいマウス対応にしてくれても，とか思いますけど。

それとアイコンの操作性は抜群によくなりました。ゲームのシステムとしてはかなりいいできですね。で，ゲーム全体としての評価なんだけど，キャラクターもかあいいし，ストーリーもスイスイ進んでくれるし，それでいて盛り上がるしで，もう最高。たぶん，いまが締め切り前でなかったなら，徹夜しようがなにしようがぜーったい一気に解きに走ってしまったと思うくらいです。オタッキーと言われようがナント言われようが，私は絶対自分で買います，これ。

熱中度▶▶▶▶▶▷▷ （て）

▶あの「第４のユニット」がX68000 に登場したんだ。以前私がハチャメチャなレビューを書いたことがあった，あれだ。ゲームの内容自体はＸ１版と変わりはないようだね。X1版のときはディスクアクセスがあると音楽のテンポが狂いに狂っていたけど，そのへんは改善されているし，曲自体もバ

ージョンアップしているみたいだ。また，68ならではのサンプリングによる効果音もなかなか。あの，特殊アイコン選択方式（？）も健在。

X1のときはカーソルの移動が重かったけれど，68版は軽い軽い。あの，チョコマカチョコマカした戦闘シーンも，まあまあ見られるものになっているし，合格かな。けれど，ソフトウェアキーボードがドバッて出てきてしまったり，68なのにマウスに対応していないのはいただけないなぁ。でも全体的に見て，遊べるものになっているのは確か。２を出すときはさっき言ったところを改善してよ。

熱中度▶▶▶▶▶▷▷ （善）

X68000用 5″2HD版２枚組 6,800円
データウェスト ☎06(968)1236

極道陣取り

隣のシマに殴り込みをかけては縄張りを広げるという，極めてアブナーイ，やくざ屋さんシミュレーションウォーなのです。

▶陣取りゲームというのは，ごく簡単なウォーゲームみたいなものである。このゲームもマップが12種類用意されていて，ゲーム画面は疑似ヘックスだったりする。しかしその上にいるのはなんと極道の面々なのだ。

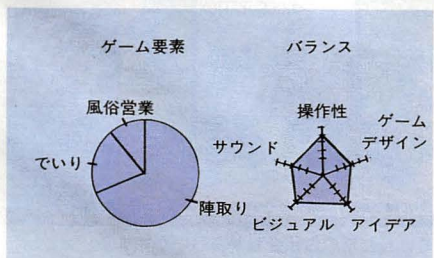
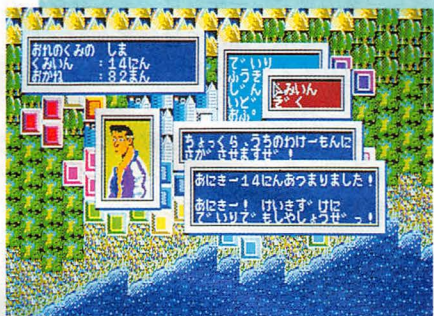
プレイヤーがすることは簡単である。資本金でキャバクラやソープランドを作り，軍資金を調達し，組員を集め，隣接するほかの組のシマに「出入り」を仕かける。たいていの場合，人員を大量に投入すれば勝負に勝てるようだ。こう簡単だとすぐに飽きてしまいそうだが，実際はそうでもない。自分のシマが広がるとすべてのシマをフォローするのが難しくなり，どうしても手薄なところができてしまう。だからいつほかの組に乗っ取られやしないかと緊張の連続なのだ。実際，奪った自分のシマの安全を確保するには，その取り方に戦略が必要とさ

れるのである。

このゲーム，単なるキワものだと思ったら大間違い。かなりプレイヤーをのめり込ませるモノを持っているのだ。

熱中度▶▶▶▶▶▷▷　(R.K.)

▶極道は金を儲ける。「ぱちんこやなら3000まんでりっぱなのができますぜぇ」。極道は人を集める。「あにきー15にんあつまりました！　あにきー！　けいきづけにでいりでもしやしょうぜっ！」。極道はでいりをする。「ヤロー調子にのってんじゃねえんだよ!!」，「死ねやザコどもブッつぶしてやる!!」，「あにきー!!　らくしょうですぜぇらくしよう!!」。負けることもある。「あにきー……やられちゃいました。きをとりなおしていきましょう！　あにきー!!」。そういうとき，極道は殺し屋を雇うこともあれば，族を仲間にすることもある。極道は身内から上納金を奪う。「あにきー！　なんわりのかねをもっていくかきめてくだせぇー」。

そんなこんなで，極道の世界をコミカルにシミュレートしてみたのがこのゲームである。マップはたくさんあって，操作性も上々，ルールも簡単，エンディングはしゃれている。ときにはこんなのも楽しい。

熱中度▶▶▶▶▷▷▷　(K)

X1/X1turbo用　5″2D版2枚組　6,800円
(2ドライブ専用)
マイクロネット　☎011(561)1370

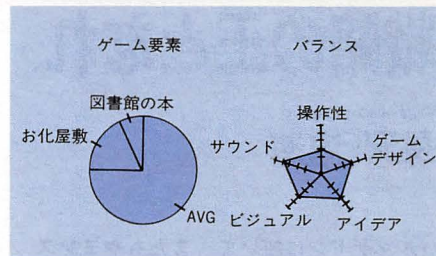
白夜物語

恐怖の館に迷い込んだグループが巻き起こす，笑いと恐怖が交錯したコミカルなAVGなのです。

▶立ち上げた瞬間からイヤな予感。まずオープニング。最近は凝ったものが多いなか，これは簡単に過ぎて，いきなり肩すかしをくらった気分です。次にグラフィックですが，ぼやけた感じを出そうと試みたのかもしれませんが，画面が小さいこともあって少々見にくくなっています。構図は結構いいと思いますし，怪物も気味悪い感じが出ているだけに残念です。それより，なんともいただけないのが，キー反応の鈍さです。画面上には常に動詞群と名詞群が表示されていて，個々より選んでコマンドするのですが，コマンドを実行させたあと，目で追える速さでいちいち名詞を書き直すのにはあきれてしまいました。また，ストーリーも特に際立ったものとはいえませんし，スケベなギャグもあまり笑えません。

とにかく，まずはお膳立て，操作環境などをしっかりしたものにしてほしかったと思います。

熱中度▶▶▶▷▷▷▷　(お)

▶最近，復活の日は近いといわれているAVGの最新作である。「白夜物語・ウインチェスター家の末裔」という題名は，最近流行の推理もののようであるが，実は「笑いと恐怖のコミカルホラーアドベンチャー」なのであった。いまではコマンド入力のAVGというのは絶滅してしまったようで，このゲームも例にもれずコマンド選択式である。しかも，このストーリー展開や画面レイアウトを見ていると，まるでファミコンをプレイしているかのような気分にさせてくれる。やはりマウスではなくて，「全機種ジョイパッド対応」というのがナニなんだろう。

で，内容なわけだが，「笑いと恐怖の……」という見事なまでのキャッチフレーズからもわかるように，誰も笑わないようなギャグの連発なのであった。このあとX68000にも登場するそうだが，もう少し笑いのセンスをパワーアップしたものであってほしいものだ。

熱中度▶▶▶▷▷▷▷　(M.Y.)

X1/X1turbo用　5″2D版2枚組　7,800円
(2ドライブ専用)
イーストキューブ　☎011(711)7709

ゲーム要素　バランス

図書館の本　お化屋敷　AVG　操作性　ゲームデザイン　サウンド　ビジュアル　アイデア

今月はX68000投稿のススメだよ

えー，皆さんも知ってのとおり，Xシリーズの最上位機種としてX68000シリーズがあって，このマシンは16ビットMPU，ハードウェアスクロール，FM音源，スプライト機能搭載と，数年前であれば，アーケードゲームでさえかなわないであろうという，もの凄いフル装備マシンなわけです。

で，当然のように，ほとんどゲームセンターのようなゲームが市販ソフトではビシバシと出てきたわけです。が，一方のピコピコゲームなどの「自分で作る小さなゲーム」のほうはあまり元気がない。その昔，ゲームを作ろうとしても，PUT命令じゃキャラクタの表示が遅すぎて「スプライトがほしいよう」と言っていたのは，確かそんなゲームを作ってた人じゃありませんでしたっけ？

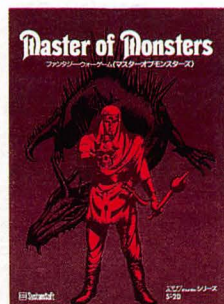
せっかく，ハードがここまでよくなったんだから，X68000時代のピコピコゲームというのを作って編集室に送ってみませんか？　この前，編集担当氏も「もっと初心者向きのX-BASICのゲームを載せたいなあ」と言ってましたことですし……。　(で)

THE SOFTOUCH

●Master of Monsters

知力と魔力で戦略の王となれ

Nagasawa Atsuhiro
長沢 淳博

ハルマゲドンに続いて，またもやモンスターが主人公のゲームの登場です。でもこちらはお馴染みシステムソフトのストラテジック。召喚してはレベルを上げて，モンスター部隊を立派に育てていきましょう。戦闘モードも見物です。

X1turbo用　5"2D版3枚組　8,000円
(model 10は要CZ-8BGR2)
システムソフト　☎092(714)6236

魔王ガイアが魔法陣の上に立ったとき，まばゆい光とともに4人の魔法使いが現れた。4人の魔法使いはそれぞれしもべであるモンスターたちを召喚して，次第に勢力を誇示するための血みどろの戦いを繰り広げようとしていた。そのモンスターたちのなかには，戦いを繰り返すうちに，さらに強力なモンスターへと成長するものたちもいた。

彼らの戦場にはいくつかの塔が存在し，その塔ではモンスターたちの魔力を上げたり体力の回復，または占拠した塔の間をワープできるなど，数々の謎の力を秘めていた。どうやらこの塔の占領が，この戦いのカギを握っているらしい。

こうして，大魔法を操るマスターとなり，自分のしもべであるモンスターたちを召喚しながら，全世界制覇の野望に燃えた戦いの幕がここに切って落とされるのだ。

＊　　＊

今回，とうとうX1turboにあのMaster of Monstersが登場です。当然これは，工場や都市を塔に，工業力や金額を魔力にアレンジしたSUPER大戦略のモンスター版だと思っていればいいのです。という，甘い言葉で巧みに受け流しつつ，皆さんをドロ沼の世界へと誘い込むのでありました。

いざ，開戦！

気温22度，快晴。温風ヒーターの風，風力1メートル。今日は絶好のMaster of Monsters日和である。この前，練習だと思って手を出した2人用マップは，どうやら難しいと見た。そこでコンピュータ同士を戦わせておいて，正々堂々と横からおもむろに侵略するという，セコイ手段がどうやらこの私には向いているらしい。この手を使って，いざ，マップ3の「EGG OF DRAGON」に出陣じゃあ。ちなみに登録されているマップは20種類。おまけにマップエディタも付いたお徳用なのです，このゲームって。

このマップ3は，俗に言う3カ国モード。赤，青，緑の3色のなかから自分の国を決めます。赤は最初から塔の保有数が多く，城は湖に囲まれていて守りが固そうなのですが，ちょっとほかの2つとは離れた場所にあるので，少し立地条件がよくなさそう。そこで青のウォーロックで始めることにしましょう。

まず，設定をすませたあと，開始のファンクションキーを押す。このようにファンクションキーで操作できるとは，ますますもって大戦略であることよ。

定石どおり，戦いは塔の奪い合いから始まるだろうから，召喚するモンスターは足の速そうなのから選ぶことにする。基本的にこのゲームでマスターは5タイプいて，それぞれウォーロック，ソーサラー，ネクロマンサー，ウィザード，サモナーに分かれている。そしてタイプ別に8種類のモンスターを大魔法のパラメータに応じて召喚することができる。また，このモンスターのなかには戦闘時にのみ召喚できるものや，2段階，3段階にレベルアップするものが含まれている，といったぐあい。

当然，敵モンスターとの相性なんかもあったりするから，安易に魔力が低いから弱い，といった判断をしているようでは，この乱世の時代は生き残れない。

で，足の速いモンスターの続きだが，うちのメンバー（ウォーロック）のなかではロック鳥とユニコーンが結構いけそう。これをちょっくら50の魔力のなかから大量召喚(?)してから，周りの様子を見てみよう。

敵陣のなかにはレイスなんかがいる。こいつには天使をぶつけたほうがいいみたい。だって，付録のモンスター総合戦闘表を見てみると，相性はいたってヨロシの2重マルが付いているんだもーん。おおっと，ここで相手が大魔法を使ってうちのロック鳥を「行動済み」にしてきたな。この大魔法とは，マスターの持つ魔力のパラメータに応じて回復や相手の行動を止める，または電撃や流星といったものまで，防御から攻撃へと全部で10種類も用意された大技なのである。だから「行動済み」には「再動」の大魔法を使って，相手の鼻をあかしてやる。

そうこうしているうちに，赤と緑が勝手に戦闘を始めたようだ。このゲームのコンピュータの思考ルーチンは，まず身近に迫ってきた敵を取り囲んでは袋ダタキ，という実にカワイイ性格を持っているらしい。おまけに「停止」の大魔法を多用することもわかってきた。そこで私はかねてからの計画どおり，このスキを縫って反対側の塔の占領に出かけることにしよう。どうやら

こうした塔の占領合戦から戦いが始まる

戦況は赤が圧倒的に有利らしいから，緑が一方的にやられてしまわないように，愛の手を差し伸べることを忘れてはいけない。とにかくいまは，打倒，赤組なのである。

いよいよ戦いも中盤戦

ターン数も11となってくれば，次第に赤組が緑の陣営を脅かし始めた。しかし，そこは，青組応援団と緑組のガンバリによって，ジリジリと赤組を押し戻さなければならない。こうして，青組の応援を得た緑組はその鉾先を赤組に向けたまま，次第に勢力を取り戻していったのであった。ここでもやはり塔の役割は大きい。赤の塔の周りにいる赤組軍団を追い払ってやって緑組に占領させれば，緑組は回復した部隊がまた再び活躍することだって可能なのだ。このあたりがこのゲームでは重要な駆け引きとなってくるに違いない。

こんな無駄話をしている間に，なんと，うちのロック鳥が敵の城の隣にある塔を占領した。こうなればしめたもの。次のターンで大魔法を使って，ここにぼこぼこと身内をワープさせていけば，遠くにいるメンバーを簡単に敵のボスにぶつけられる。

このほかにもこのゲームには，地形効果や時間効果といったものもあって，陸や海によって行動範囲が違っていたり，朝・昼・夜のなかで攻撃力が違ったりと，なかなか細かいところまで考えられているようだ。

さて，最終ラウンド

いよいよ戦いも終盤戦。我が青組がマップの70%を占めるようになってくると，緑も赤も塔はほとんど持っていない，という状況になってきた。これでは大魔法を使ってもモンスターたちを召喚できないので，宝の持ち腐れちゅうことになってしまう。しかし，それですんなり黙っている連中ではない。マップ中に20個もの隕石をぶつけてくるという「流星」の魔法をばしばし使ってくる。ホントに，いやなやつ。この魔法は，場所を選ばずマップ全体に向けてで

彼らが大魔法によって召換できる面子です

「行動済み」には「再動」で対抗

たらめに攻撃してくるから，なかなかにスリリング。近くに命中すると，おっと次はそろそろ危ないぞ，なんて思いきり盛り上がってしまう。

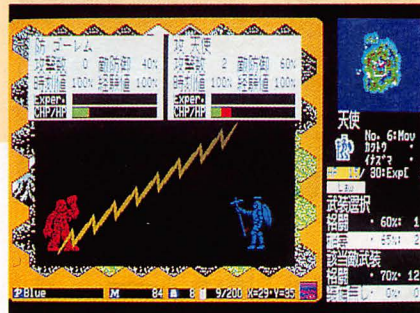
現在の戦況は，赤組と緑組の2つを自分が相手にしているわけだから，完全に2対1で戦っている。そのうえ，ちょっと目を離したスキに緑組がサーペントなるモンスターを持ち込んできた。こいつは滅法強いから，どんどん塔を奪取されてしまう。仕方がないから，こっちもサーペントを呼び出して対抗だ。こうなりゃ，鬼ごっこみたいなもの。ここで取られたら取り返すゾ，と熱くなってはいけない。一点に気を取られているうちにシッペ返しを喰わされる，というのはよくある話。ひとまず，動かざること山のごとし，である。

結局，一歩下がって塔を明け渡しておいて様子を見ることにした。本当はモンスターたちをばしばし召喚して数で押すというのも考えたのだが，せっかく成長型のモンスターがいるのだから，この連中をフルに使って成長を見てみることにした。すると天使が大天使へ，ロック鳥は火の鳥へと，まあ，ご立派に育つこと。当然，レベルアップもされるから，戦いもずいぶん楽になった。

そうして少し遊んだあとは，サーペントを近くの自分の塔のなかに待機させて，そこを拠点に，攻撃してはまた戻るというヒットアンドウェイ戦法に出た。敵のマスター自身のヒットポイントは回復することはないから，こうして少しずつじりじりと攻めていけば，もう結果は見えてきたようなもの。我ながら，なんと賢い戦法よのう。

そしてチェックメイト

もう機は熟した。あんなに最初がんばっていた赤組も緑組も塔の数は2つずつしかなくなった。おまけにそれぞれのマスターのパラメータも残りわずか。さて，まずは赤から料理するか。赤組マスターの上からは火の鳥，下からは大天使をぶつけよう。

ちょっくら緑組の応援に出動

これが天使のイナズマ攻撃なのだ

緑のほうにはドラゴンの大部隊で周りを固めておこう。

いきなり，赤のマスターであるネクロマンサーに大天使が襲いかかる。そして火の鳥がとどめを刺した。こうしてあっけなく，赤組さんは我が軍に敗北した。この勢いで緑組も落としてしまおう。まずドラゴンの先兵部隊が敵陣に乱入，緑組のマスターであるソーサラーに1回ヒット。続いてユニコーンも1回ヒット。ここで赤組を倒した張本人，火の鳥が到着するのを待って再び攻撃だ。そして火の鳥の攻撃，バシッ。一撃でエンド。グラフィック画面が出たあと，勝ってよかったねとメッセージが出ておしまい。実に爽快な気分である。

戦い終わって日が暮れて

このゲームには，こういった単発でプレイできるゲームのほか，勝ち抜き戦形式でレベルアップしたモンスターを連れて行ける「キャンペーンモード」とか，「マップエディタ」や「ユーティリティ」，さらにゲームを難しくして遊ぼうという方には「索敵モード」など，いろいろなモードもあって楽しめます。

とにかくRPGとストラテジーを組み合わせた発想は，8ビットマシンでも十分に楽しめるし，特にキャンペーンモードは，このタイプのゲームにはピッタリだと思うんだけど。そうか，大戦略にも付けてくれればいいのか，そうすればもっと遊べそう。いずれにしてもこのゲーム，自称「戦略家」の私からのお勧め品です。

●サンダーフォースII

地獄の横スクロール マル秘攻略法

Shimizu Kazuto
清水 和人

「ゲームディスクと目薬をセットで送ってね」と編集室に電話した彼は，その後右手にジョイスティック，左手に目薬を持つ日々を何日も続けたらしい。こうしてOh!Xの読者だけに知らされる横スクロール攻略法が無事完成に至ったのだ。

X68000用　5"2HD版2枚組　9,800円
テクノソフト　☎0956(33)5555

というわけで，今月も私のX68000は依然「サンダーフォースII」のディスクを飲み込んだままである。かつてX1のゼビウスを買ったときに付いてきた，名作ジョイスティックも真夜中の4畳半に怪しい光を放っている。そして際限なく繰り返されるステージ1から4までのデモンストレーションも賑やかだ。

夜はさらに深まりつつあった。いざ立とう戦士たちよ。缶入りウーロン茶をこじ開け，目薬の用意も万全に，長い戦いへと旅立つのであった。さて今回は，8方向スクロールを卒業した諸君に贈る，横スクロール攻略法なのである。

横スクロールの登竜門ステージ1

さて，いよいよ横スクロールの攻略の初歩にとりかかろう。ここでは，いままでのシューティングゲームと比べてもかなり高度なテクが必要とされる。最初は自分の位置がうまく把握できなくて，うまく操作できず自滅の繰り返しである。しかし，一度失敗したら2度目は絶対になにか工夫する，これがリアルタイムゲームの基本である。

普通のシューティングゲームだと，「あまり動かずに戦う」とか「ジョイスティックを小刻みに動かす」などのポイントを見つける努力をすれば，高速スクロールでも大丈夫だろう。しかし，このゲームにはそれを超えた難しさがある。沙羅曼蛇の横スクロールなどとは違い，敵の攻撃は情け容赦がないのだ。これに対抗するにはもはやパターンを覚えるしかない。敵の出現位置やオプションの出る位置を覚えて，先手必勝である。

たとえばステージ1だと，最初に赤い敵を後方中央から撃ち，下に落ちた青いやつを地上すれすれから撃破，また中央に戻って赤いのを撃つといったぐあいである。次の中型の敵がWAVE SHOTを残していくので拾って装備，すぐに撃ちながら上部のトンネルに入る。トンネル内部ではBACK FIREを撃ちまくり，トンネルが終わるとすぐさま前方上に移動して次のトンネルに備える。そしてさらにそこから抜け出るとSIDE WINDERが取れるが，WAVE SHOTのまま前進し，下のバリアを突き破り，上下にこまめに移動してバリアの向こう側の敵を撃破する。

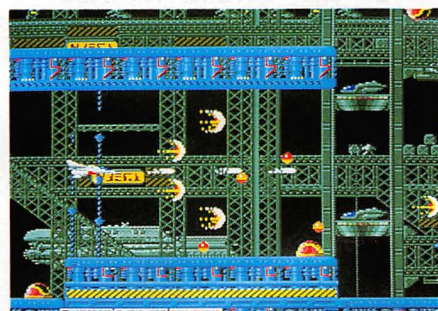
そのあと上に上がって待っていると，最初の巨大戦艦の登場だ。ここでひるんではいけない。敵は前方の上部にある青い部分からレーザーを撃ってくるが，これを攻撃が始まる前に破壊する。もしこれに失敗しても，レーザーが発射された瞬間に上下に移動し，レーザーを避けながら攻撃すれば破壊できる。

このようにして最後まで行ったら，このステージのボスキャラとご対面だ。後方の上か下で敵の攻撃をかわしつつ，連射を続ける。これを3～4回繰り返すと敵レーザーは破壊され，今度は赤い弾がバラバラと発射される。これには正面で静止したまま攻撃するのがいい。ただ，敵が移動をやめたら突進してくるので，このときは上下に素早く逃げなければならない。

こうしてステージ1は楽々クリアできるのだった。

すでにここから要注意

ステージ2は，最後のボスキャラは弱いくせに，やたら中盤がキツイ。ここでは旋回を繰り返しながら攻撃を続け，ビュンビュン飛んでくるレーザーの間をすり抜けるのには，例の小刻みなジョイスティック操作が必要だ。そのあとワープして敵が出現したり，息もつかせぬ青い球の攻撃が繰り広げられたりと，一瞬パニックとなる場所がある。ここでは敵を攻撃したあと思いきって前方に出る。そうでないと後ろから壁がどんどん迫ってくるのだ。これを知らないといままでの苦労が水の泡となってしまう。

ふー，とにかく最初はなにが起きるかわからないので，とても精神力を消費してしまうのだ。

ステージ1はほんの小手調べ

8方向スクロールではマップ確認が先決

なんだコリャのステージ3と4

えーと，とにかく大変なんだ，これが。ステージ3は上中下と3つの青いトーチカみたいなのを連射で撃破したあと，BACK FIREで上の青いのを攻撃，そのあとすかさず上下のトーチカをやっつけて，WAVE SHOTを取り地上のロボットキャラを攻撃，そのあと天井から黄色いのが降ってきて地上から攻撃してくるので，コイツを地面に落ちる寸前で撃つ。次にたくさんの青い敵が突然現れて，トーチカとこいつのツープラトンの攻撃をくう。これをかわすと今度はエレベーターと動く壁が出現するからすり抜けて，最後のイモ虫キャラとの対決となる。

一気にしゃべると，ステージ3はこうなっている。これがビュンビュンと目の前に現れては攻撃してくるので，とにかく自分の目で確かめてほしい。これじゃ，命がいくつあっても足りんワイ。

しかし，ステージ4はもっととんでもない。最初すぐに赤い繁殖型のキャラが画面上を埋め尽くす。しかし，こちらも横スクロールに入ったばかりの一瞬の間は無敵状態だから，すかさず体当たりやBACK FIREであっさり全滅させて，まずはストレス解消。続いて後方やや下寄りで攻撃していると，WAVE SHOTが手に入る。そのあとすぐに凄い数の敵が押し寄せてくるので，すぐさまWAVE SHOTに切り換えて撃退。そうしてATOMICやMEGA FLASHを拾いながら，警報が鳴り始めたところで下のトーチカをやっつけて，今度はBACK FIREで青いのとかヘビみたいなのを攻撃。ここで，「あー，しんど」などと一瞬でも油断したら，ロボットの餌食にされるので，下で事前に迎え撃つ。このあとの小型艦でちょっとだけ遊んで，次に備える。

いよいよ超美技を見せるときがくる。前方ちょい上で巨大戦艦を連射，このチャンスを逃すと四隅から敵の弾が襲いかかるのでそれを画面中央でかわす。ここでHANTERを気持ちよく連射しながら上からのツララを前後に避ける。再び警報が鳴ったら下のトーチカを攻撃。さあ，次にいよいよ巨大戦艦出現とともに敵の雨あられの攻撃が始まる。「おっ，これでついに最後か」と思っているとなんとか戦艦が通りすぎてくれたので，「じゃ，終わりかナ」と思うとまだまだこのステージは続く。触覚の大きいヤツみたいなのをやっつけて進むとどんどん通路の上下が狭くなるわ，上からは粘着質の滴みたいなのが落ちてくるわで大騒ぎ。その先にいるヘンな生物みたいなボスキャラを，後方中央よりちょい前で滴からの安全を確保しつつ攻撃する。敵の口から発射される弾はちょっとよければ簡単に逃げられる。しかしここで浮かれていると，次のステージ5で地獄を見るハメになる。

まずは左が恐怖のステージ5。これの高速スクロールはまるでジェットコースター気分。下はステージ4の触覚みたいなやつと、そのお隣がなにが起きるかお楽しみの，最終ステージなのです

勝手にしやがれのステージ5

ここまで来ると，心底疲れるのでブレイクキーでポーズにして，ウーロン茶でも飲んで，肩の力を抜いてひと休み。さあ，いよいよ横スクロールは，これが最終ステージでっせ。

むむ，いきなり警報が鳴っているぞ。おっと，巨大戦艦が登場。これはエゲツない攻撃。中央ちょいで連射して，すぐに下に移動してトーチカを破壊。MEGA FLASHにチェンジして高速スクロールが始まってもいいように準備。ドワーッ，と画面が流れ始めても落ち着いて赤い敵をかわす。それが終わるとステージ2にいた避けなきゃいけない戦艦がヌッ出てくる。はいはい，わかりましたでぇ。コイツは画面内をクルッと回ってちょちょいのちょい。あら，やられてもうた，こら，あかんわ。

気を取り直して迷路を進む。ここでは早くATOMICを取らないと通れない。壁に穴を開けたら，LASER取ってWAVE SHOT取って，と思っていたらいきなり逆スクロールしてUターン。次は巨大戦艦の挟み打ちにあうから，画面中央からLASER攻撃。そのあとちょい前に出て下のトーチカと上のトーチカやっつけて，上の通路をくぐり抜けてロボットたたいて，上のトーチカやって，壁のまん中掘り進む（こりゃ，忙しいや）。

そうして赤い壁を入ると突然中央にトーチカが出てくるから，コイツを後方から撃つとまたまた高速スクロール。最初の分岐は下へ，次からは上を選ぶと無事ボスキャラにご対面。あー，疲れたー。

ステージ6にはなにが待つ

死ぬほどしんどい横スクロールを抜けると，最後のステージ6が待ち受けている。ここは8方向スクロール＋αなのだが，なにしろ敵基地以外はすべて自動追尾の連射型トーチカ。避けるためにはほとんどマラドーナのドリブルがごときテクニックを必要とする。「あー，もうわていやや。もう勝手にしておくんなまし」と試合放棄したふりして，またもや薬局で目薬を買いあさってしまう私なのであった。

ここから先は，各自がんばってみるように。このサンダーフォースIIの誠の姿がここに隠されているのだ。

THE SOFTOUCH

SOFTOUCH PRO-68K

パックマニア
ボスコニアン
パワーリーグ
めぞん一刻・完結編
第4のユニット
三国志
蒼き狼と白き牝鹿・ジンギスカン
ウォーニング
ヒストリーガイア
WINGS OF FURY
ファンタジーⅢ・ニカデモスの怒り
口説き方教えます

伊東　建文　(20)　神奈川県

これが新しい殺人倶楽部。このオープニングは，琥珀色の遺言と並ぶほどの素晴らしさなのです

先月，このコーナーでもご紹介した，X68000用殺人倶楽部が完成間近のようです。今月号が発売されるころには，もう店頭に並んでいるかもしれません。とにかくこのリバーヒルのX68000用AVGはマンレク以来，ますますそのグラフィックと演出効果のうまさには磨きがかかってきました。ゲームの見せ方という点では，ほとんど芸術の枠に達したといってもいいくらいの仕上がりです。

そういえば，前にご紹介したガルフォースや，今月ご紹介のめぞん一刻とともに当分の間はシューティングとAVGが主流となりそう。しかし，それに負けじと海外からの移植ものRPG，M&Mやファンタジー，そしてウルティマシリーズ(Ⅰ～Ⅳ)が続きます。さて1989年のゲーム界に，X68000はどのような衝撃を与えてくれるのでしょうか。

X68000ソフト&ツールズ

☆……11月30日現在発売中　★……近日発売予定

★パックマニア

パックマンシリーズが出てもう10年ほどになろうか。元祖パックマン，ミズパックマン，パックマンジュニアなどこのシリーズは種類も多い。そして，このなかでいちばん新しいパックマンシリーズがこのパックマニアだ。ゲームセンターなどで馴染みの諸君も多いだろう。このリアルタイムアクションゲームは例によって気まぐれ，追いかけ，おとぼけといったモンスターをかわしながら，エサを取ってはパワーアップしてボーナス点を稼ぐところは元祖と同じ。パックマニアではそれに加えて，画面構成が立体的になり飛び逃げやスピードエサなどの要素が増えた。とにかく，音楽も画面も綺麗で楽しめるゲームとしての完成を期待したい。

X68000用　5″2HD版　価格未定
シャープ　☎03(260)1161

★ボスコニアン

電波新聞がドラスピに続くアーケードからの移植ゲームは，ナントあの懐かしのボスコニアン。かつて，一世を風靡した「BLAST OFF！」の音声，自分を常に中心に置き，広い宇宙空間を駆け巡る8方向スクロール，そして，その向こうにそびえ立つ敵の基地，前後どちらにも弾の出せる自機。このどれをとっても懐かしい。しかし，これがX68000版ともなればきっとなにかが変わっての登場に違いない。どのような新しい工夫が加えられるか期待したいところだ。

X68000用　5″2HD版　価格未定
電波新聞社　☎03(445)6111

☆パワーリーグ

お馴染み，ハドソンの野球ゲーム，パワーリーグが登場だ。このゲーム，Mr.プロ野球などのプレイヤーがオーナーになるタイプの野球ゲームではなく，プレイヤーが選手となって実際にプレイするタイプの野球ゲームである。リアルタイプのキャラクターたちが，投げ，打ち，フィールドを駆け巡りながら熱戦を展開する。画面構成は，ピッチャー・バッターをメインにした3Dモードと，球を打ったあとのフィールドモードの2面構成で，野手のフィールディングも自分で行え，ダイビングキャッチや場外ホームランなど数々のプレイが楽しめる。

X68000用　5″2HD版　7,800円
ハドソン　☎011(841)4622

★めぞん一刻・完結編

めぞんファン待望のX68000版がようやく発売されることになった。このゲームのストーリーは基本的にはまったく同じで，五代君がなかなか管理人さんに思いを打ち明けることができないまま，一刻館の住人たちにふり回されながらも孤軍奮闘するというもの。そうして，あの思わず涙してしまいそうなラストシーンへとプレイヤーを誘ってくれる。特にX68000版は無駄な会話のやりとりを改善したり，グラフィックを全面的に描き直しての登場。このゲームは，プレイしたあと，少しだけハッピーな気分にさせてくれるAVGといえそうだ。

X68000用　5″2HD版5枚組　9,800円
マイクロキャビン　☎0593(51)6482

☆第4のユニット

東京は練馬の上空で謎の航空機撃墜事件が起きた。国籍そのほか事故機の内容は一切わからなかった。その翌日，優介は公園でかわいい女の子を助ける。彼女は記憶を失ってしまい，かすかに覚えているのは「ブロンウイン」という言葉だけ。彼女はいったいなんのためそこにいたのかはおろか，自分の名前さえも思い出せない状態だった。優介

パワーリーグ

は彼女の記憶を取り戻すべく動き出すのだが，そこで次第に事件とこのタイトルである「第4のユニット」の全貌が明らかにされていくという，お馴染みデータウエストのAVG。プロレス技を多用した戦闘場面には笑える。追って続編の「第4のユニット2」(7,600円)もこの12月には発売の予定だ。

X68000用　5″2HD版2枚組　6,800円
データウエスト　☎06(968)2792

★三国志

X1などでファンも多い，三国志がX68000にも登場する。舞台は1800年ほど前の中国大陸。年代別に分かれた5つのシナリオをもとに，それぞれ董卓打倒，曹操の台頭，新時代の幕明け，孔明の出盧，三国の時代，と名づけられたストーリーが展開される。それぞれのシナリオは実在の「三国志演義」をもとに作成されているため，中国大陸を舞台とした歴史の世界を楽しみながら学ぶこともできる。中国大陸58カ国を，果たしてあなたは無事統一できるだろうか。

X68000用　5″2HD版2枚組　14,800円
光栄　☎044(61)6861

★蒼き狼と白き牝鹿・ジンギスカン

光栄の歴史シミュレーションゲーム3部作から，三国志に続き「蒼き狼と白き牝鹿・ジンギスカン」も発売される。このジンギスカンは，12世紀も半ばのユーラシア大陸を舞台に，モンゴル帝国のジンギスカン，イングランドのリチャード一世，ビザンツ帝国のアレクシオス，日本の源頼朝といった歴史的英雄となり，あの広大なユーラシア大陸に空前の大帝国を築くべく侵攻を開始する。シナリオはモンゴル編と世界編の2つで，戦闘で勝利し国を支配しては，政策や人員配置を決定するなど，細かい設定まで気を配りながら，自分だけの一大帝国を築く楽しみが味わえる。

X68000用　5″2HD版3枚組(予定)
9,800円(予価)
ミュージックテープ付きサウンドウェア
12,800円(予価)
光栄　☎044(61)6861

★ウォーニング

SFウォーシミュレーションゲーム，ウォーニング。このゲームは，大小7つの惑星そしてひとつの人工惑星からなるプシェード星系が舞台。ここではそれぞれの星が独立した国家を築いていて，盛んに貿易が行われている。ここでプレイヤーはトレイダー(貿易商人)となり，交易を拡大し利益を上げつつ，さらにはバンデッドと呼ばれる宇宙海賊の襲撃もかわしながら行方不明になった兄を捜し出すのだ。なお，このX68000版には，オリジナル曲の追加や禁断のアイテムの追加など，他機種のものよりさらにグレードアップされての登場だ。

X68000用　5″2HD版2枚組　7,800円
コスモス・コンピューター　☎03(770)1821

★ヒストリーガイア

いまは昔，ガイアの大陸に栄えていた4つの国のひとつ，ヴァルゼリアの国王ボーゼル一世が魔術師ベルナリッヒより「闇の術」を体得したとき，4国の均衡は崩れ始めた。ボーゼルはその力によって大陸のすべてを彼の手中に収めようと動き出した。しかし，平和を乱す影魔術に光魔術が対抗すべく立ち上がった。それは魔術師バーリンにより「光の術」を授かったエルスリードの国王ジークヘルマンである。こうして大陸の覇権をかけて，エルスリードとヴァルゼリアの2大国の長く激しい戦いの日々が始まった。あのX1でお馴染みのシミュレーションゲーム，エルスリードが，思考ルーチンの強化や美しいグラフィックとともにX68000に帰ってきた。

三国志

ウォーニング

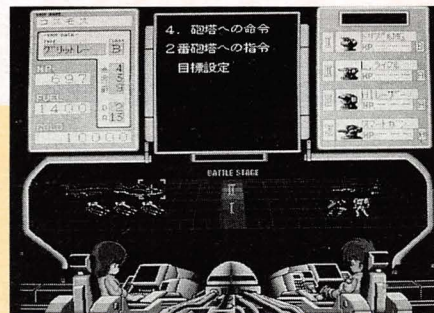
X68000用　5″2HD版　価格未定
日本コンピューターシステム　☎03(486)6311

★WINGS OF FURY

ブローダーバンドジャパンからフライトシミュレータ型戦略ゲーム「WINGS OF FURY」が発売される。このゲームは1940年，太平洋戦争真っただなか，機銃，ロケットランチャ，対艦魚雷，爆弾などの武器を搭載したゼロ戦のパイロットとなり，敵迎撃機とドッグファイトを行いながら敵基地や敵戦艦撃破といった作戦行動を展開する，フライトシミュレーションゲームだ。画面構成は上空からの俯瞰モード，降下時のフライトシミュレータモードに分かれ，それらの切り換えはすべてオートマチックで行われる。また，上昇，下降時そして旋回時のリアルなアニメーションは，これまでのフライトシミュレータでは見られないほど，忠実に再現されているらしい。X68000シリーズの機能をフルに生かした巨大戦艦の動き，敵戦闘機のリアルな移動など，実に見せ場の多いゲームといえそうだ。

X68000用　5″2HD版　価格未定
ブローダーバンドジャパン　☎03(341)1131

☆ファンタジーIII・ニカデモスの怒り

なんと，X68000のファンタジーシリーズは，いきなりIIIからの登場。このファンタジーIIIは，ファンタジーシリーズの完結編に当たるもので，IとIIの舞台となったジェルノア島，フェロンラ島からそれほど離れていない場所に位置するスカンドール島を舞台に物語は始まる。そうして復讐の鬼と化して前作までの勇者たちを待ち受けるニカデモスを相手に，今度という今度はニカデモスの息の根を止めようと立ち向かう勇者たちの熾烈な戦いが繰り広げられていく。次々と襲いかかるモンスターを撃ち破り，スクロールに隠されたメッセージにそって，島の奥深くに潜むニカデモスを追い詰めた先には，誰も見たこともないような壮絶な戦いが待ち受ける。

X68000用　5″2HD版3枚組　9,800円
スタークラフト　☎03(998)2988

☆口説き方教えます

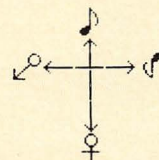
X68000シリーズにもハードのあのシリーズが登場した。その第1作目がこの「口説き方教えます」。作者いわく，このゲームは「口説きのプロ」の指導のもとに作成されたもので，彼は「世の中の女性を喜ばせる」という信念と経験に基づき，長年築き上げてきたものをいまこのゲームに託したそうで，なんでも，自称，「知らず知らずのうちに女性を喜ばす方法を身に付けていくことができるゲーム」ということらしい。あの筋のX68000ユーザーの方にお勧め？

X68000用　5″2HD版　6,800円
ハード　☎03(837)1893

使って楽しいHandy Print Jack

最近では，パーソナルワープロのほうがパソコンより先に面白い小物を用意するようになってきた。それは，はなはだ面白くないというわけか，書院用のハンディプリンタをX68000に流用してしまおうというのが，この「Handy Print Jack」である。インクリボンを2回りほど大きくしたような書院用ハンディプリンタ（WD-01PP)を，Z'sSTAFF PRO-68Kのプロテクトモジュールのような箱を介してジョイスティック端子につなぐ。きっと電源はジョイスティック端子からとれる5Vだけだろうなぁ，と思うといじましい。

印字はというと，専用のツールを使って挑戦したが，熱転写であるからそうきれいではない。が，リボン以上の幅の字は打てないとはいえ，手軽にどこにでもワープロ文字が書けるというのは魅力だろう。付属のツールはいまひとつだが，ドライバとCの関数なんかも公開されているので，自分で作れる人にはいいオモチャかもしれない。しかし，シャープのハンディプリンタ(取扱説明書付き！)とこのソフトで2万円以上というのは割高なので，もっと安くするか，もう少しいいツールを付けてほしかったとも思う。ともあれ，ビデオのレーベルやらノートの表紙やら，普通のプリンタではできないことができるのは確かである（その分，普通のプリンタ代わりにはならないけどね)。　(K)

Handy Print Jack　24,800円
計測技研　☎0286(22)9811

愛の十字架

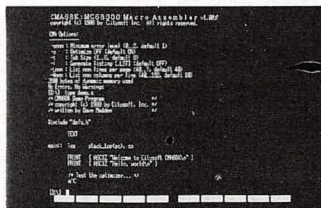
印字サンプル(縮小率60%)

THE SOFTOUCH

●マクロアセンブラCMA68K

充実の開発環境 新アセンブラ登場

Nakamori Akira
中森　章

今月はアセンブラCMA68Kの登場です。X68000の発売から1年半，最近になってようやく整い始めた開発ツールは，いったいどのようなスタイルをしているのでしょうか。まずは，そのあたりからじっくりと見ていくことにしましょう。

X68000用　5″2HD版　29,800円
シティソフト　☎06(927)1060

歓迎，新マクロアセンブラ

日本で一番進んだゲームマシン，これがX68000に対する現在までの一般的な見方なのかもしれません。実際のユーザーである私たちからすれば，目立っている部分だけでレッテルを張られてしまうことに不満がない訳ではありませんが，そうかといってゲームをプレイするうえで最高のマシンであるという評価は，嬉しいのも事実です。

しかし，パソコンの楽しみはゲームだけではありません。自分でプログラムを書き，自分でマシンを思いどおりに操る楽しみがあるはずです。X68000の発売当初，市販ソフトがほとんどない状態にあってもX68000が高い人気を保っていた理由のひとつは，アセンブラとリンカが標準装備されていて，自分自身でプログラムを作れる環境が用意されていたことでしょう。先月の12月号の斎藤晋氏の記事によると，X68000用のC compiler PRO-68Kが1年間で7千本も売れたそうですが，これを見ても多くのユーザーが自分自身でプログラムを作ることを望んでいるということがわかります。

現在，X68000には標準的なCコンパイラ(XC)と，標準的なアセンブラ(AS.X)がシャープから発売されていますが，このような状況下にあって，より高機能でより使いやすいコンパイラやアセンブラがほかのソフトハウスから発売されてくることは歓迎すべき状況です。その第1弾として先月号ではプリプロセッサPP68Kを紹介したわけですが，今回はシティソフトから発売されたマクロアセンブラ「CMA68K」を紹介することにしましょう。それとここで前もってお断りしておきますが，今回のレポートに使用したCMA68Kは完成直前のバージョンのため，市販品とは一部仕様が異なる場合があるかもしれません。

また，このCMA68Kを使うにあたっては，後日シティソフトから発売予定のSLINK，または福袋のLK.Xのどちらかのリンカが必要ですのでご注意ください。それでは，その実態を探ってみることにしましょう。

CMA68Kの新たな機能

まずは，このCMA68Kが私たちにどのような開発環境を提供してくれるか，特徴的な部分をいくつかピックアップしながら見ていくことにしましょう。

CMA68Kのアセンブリ言語の記述は，AS.Xのスーパーセットになっていて，AS.Xの文法にいくつかの疑似命令，プリプロセッサ命令，オートコードジェネレーションが追加されたものになっています。このため，AS.Xによって開発したプログラムはすべてアセンブルすることができるようになっています。以下にこれらの新機能について簡単に説明します。

まず，新たに加えられている疑似命令は，

1) align
2) rept～endrept
3) .inform
4) .warn
5) .error
6) ascii/asciz

の6つ，またプリプロセッサにおいては，

1) #include
2) #define
3) #undef
4) #line
5) #if
6) #ifdef/#ifndef
7) #else
8) #endif

などが目新しいと言えそうです。各命令についての細かい機能解説については省略させていただきますが，これらの命令のなかで実際に使ってみて感じたことがあります。それはこのCMA68Kでは，プリプロセッサの機能の一部はアセンブラの疑似命令とまったく同等に扱われていることで，たとえば，

#include → include
#define → equ (set)
#define(引数付き)　→ .macro

などといったところですが，ここでひとつ困ったことが起きてきます。それは，#defineとequを同一視することです。#defineはその場所よりあとのアセンブルに対して定義が有効ですが，equはファイルすべてで有効な定義です(したがってsetと違い再定義できない)。CMA68Kではequで定義したシンボルを定義より前で使うとエラーになってしまいます。Cコンパイラではlink命令などの引数をあとからequで定義することが多いので，バージョンアップの機会にはぜひとも修正してほしい点です。

次にオートコードジェネレーション機能ですが，このオートコードジェネレーションは，なかなかユニークな機能と言えます。アセンブリ言語のプログラムでは一度しか参照しないようなラベルを使うことがよくあります(たとえばメッセージを出力する場合のメッセージが格納されたアドレス)。このようなラベルをいちいち定義するのは面倒ですから，ラベルの内容（文字列の定義や一時的なサブルーチンの内容）を記述することで，そのラベル（アドレス）を自動生成する機能がオートコードジェネレーションです。たとえば，

```
    lea        [dc.w     234],a1
```

というのがそれで，234というデータが格納されているアドレスを自動的に生成します。もっと複雑な例としてはサブルーチンのオートコードジェネレーションで，

```
    bsr  [ pea  [asciz  "Oh!X¥r¥n"]
           dc.w        _PRINT
           addq.l  #4,sp
           rts                  ]
```

というものです。一度しか呼ばれないサブルーチンは，このように記述したほうがわかりやすいかもしれません。この例ではオートコードジェネレーションが入れ子になっています。CMA68Kの出力するオブジェクトはAS.Xのオブジェクトと互換性があります(LK.Xでリンクできる)が，このオートコードジェネレーションを使用する場合は後日発売される専用リンカを用いないとオブジェクトを正しくリンクできないので注意が必要です。

AS.Xとの特徴的な相違点

先にも述べましたが，CMA68Kは基本的にはAS.Xの文法のスーパーセットですが，細かいところでAS.Xとの相違が見られます。そのいくつかをここに示しましょう。

1) 構文チェックの厳密さ

AS.Xは1行内で命令を記述した右側にはなにを書いても注釈として無視されましたが，CMA68Kでは関係ないことを命令の右側に書くとエラーになってしまいます。注釈の前にはちゃんと「；」(セミコロン)を付けなければなりません。

2) さまざまなアドレッシング表記

ディスプレイスメント付きのアドレッシングモードの表記にはいろいろな方法があります。たとえば，

```
    4(a0)
    (a0)4
    (4,a0)
    (a0,4)
```

すべてレジスタa0に対してディスプレイスメント4の位置を示すアドレッシングモードの表記です。

3) 命令の置き換えと最適化

CMA68Kは命令コードを最適化してくれます。AS.Xでもディスティネーションがアドレスレジスタの場合はmoveをmoveaに変えたり，addをaddaに変えたり，分岐命令のディスプレイスメント長を最短になるような最適化がなされていましたが，CMA68Kはもう少し進んで，レジスタリストで指定されるレジスタの個数が1個の場合はmovemをmoveへ置き換えたりしています。ただし，AS.Xに見られるような，

```
    clr.l a0  →  suba.l  a0,a0
```

という置き換えはもうしていないようです。

4) レジスタR0～R15

AS.XではデータレジスタをR0～R7，アドレスレジスタをR8～R15といった別名で記述することができますが，CMA68Kではサポートされていないようです。コンパイラを作る人にとってはデータレジスタとアドレスレジスタを同等に扱うと都合のいい場合があるので，AS.Xはこのような記述ができるのでしょうが，Cコンパイラを大いに意識したCMA68Kにできないのは不思議です。

CMA68Kの価値

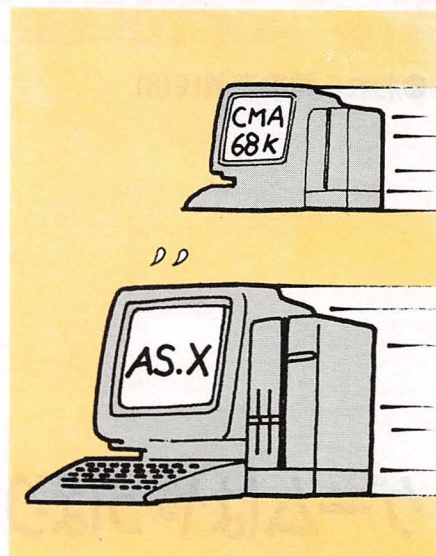
「なんで，あなたはここにいるの？」

「なんでって，自分はこのパソコンのアセンブラよ」

「大人の言いそうなことね。私が聞きたいのはそんなことじゃなくって，AS.Xとの関係よ」

「関係って，私の尊敬できる先輩よ」

*　　　　　　*

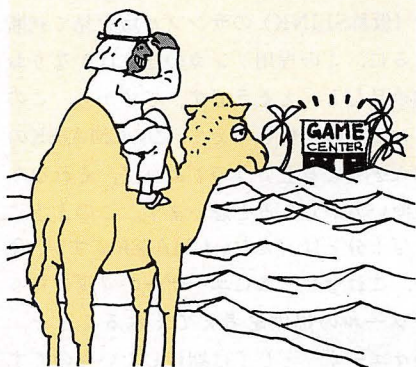
X68000にはシャープが標準的なアセンブラとして提供しているアセンブラ（AS.X）があります。そこへもってきて，このCMA68Kなのですが，AS.Xに置き変わるべき新アセンブラとして発売されるからにはそれなりの特徴を備えていなければなりません。が，私の見たところ，これまでに説明してきたCMA68Kの特徴が，AS.Xをやめてまで使うほどの魅力となるかはわかりませんでした。

このCMA68Kの特徴は，どちらかと言えば「C言語ライクなプリプロセッサ機能とオートジェネレーション機能」なのですが，しかし，プリプロセッサ機能のほとんどがAS.Xで代用できますし，いざとなったらXCのプリプロセッサだって使うことも可能なのです。

オートコードジェネレーション機能にしても，プログラムは確かに読みやすくはなりますが，記述のたびにコードが生成され，しかもそのコードは，ほかの場所からは参照できないなどといったメモリ効率の悪さが生じそうです。それに，アセンブル速度の遅さには否定できないものがあります。試しに，以前私がX-BASICの連載で作った自動作曲プログラムをAS.XとCMA68Kでアセンブルしてみると，アセンブルリスト出力をしない場合で約2倍，アセンブルリスト出力をする場合で約4倍もの時間がかかってしまったのです。

しかし，現在開発中のCMA68K専用リンカ（仮称SLINK）のサンプル版を見て判断するに，この専用リンカのほうはかなりお勧め品だと言えそうです。ですから，このリンカとの相性を考えて，このCMA68Kの購入を考えるというのであれば，それなりの使い方ができると思います。このように，まだ十分とは言えないCMA68Kですが，今後，これをベースにユーザーの立場に立ったツールの提供を考えてくれることを，我々ユーザーとしては期待したいものです。

ゲームはやっぱり心の鏡なんです

Komura Satoshi
古村　聡

ゲームが楽しめるという背景には，個人個人の経験によって培われた要素というものが影響しているはずなのです。そこで今回は，(で)氏の体験をもとに，アニメやマンガが与えてくれる要素とはなにかを，見ていくことにしましょう。

私の言い訳

私がドジをやるのは，毎月，毎月，毎月，毎月（えーい！　ここまで強調すると腹の立つ）のことなのですが，先月号でも私はおばかなことをやってしまいました。私はGAME　REVIEWの囲み記事のなかで，「ゴーバンさんの苗字は出てこない」などと書いてしまいましたが，実はイースⅠで出てました。ダームの塔の入り口を塞いでいる盗賊のゴーバンさんは，フィーナを保護した婆さん「ジェバ・トバ」(この人の名前は塔のなかで誰かが言っていた）の息子なんですから，当然「ゴーバン・トバ」なのです(ああ，なんて似合わない名前)。どうも，すいませんでした。

で，いまごろになってなんでこんなことがわかったかというと，実は，つい最近セガ用のイースが安かったので，思わず買ってしまい，私はまたもやイースを初めっからやってしまったのです。それからというもの，いまごろになってイースのマニュアルを読み返したり，神官の家系図を作ってしまったりと，すっかりイースオタッキーと化してしまいました。

実を言うと，イースを2度やって2度目は1桁の時間で制覇してしまうという離れ技をやってのけた編集室の某氏（斎藤晋さん）とは違って，私はパソコン版でやったときは4日で終わらせたくせに，セガ版では3週間たったいまでもまだダルク・ファクトを倒せてません。ちなみにこの原稿を書き始めるまでイースをやって，その前は自作のPSGドライバでイースの音楽のコピーを作ってたという，はっきり言って3週間イースにどっぷりの生活です。締め切り時間まであとわずか。うーっ，白いワニがくるよーっ！（ああ，また，埼玉の江口寿史とか言われてしまう……)。

まあ，それはともかく，なぜここまでイースというゲームは，私をのめり込ませてくれるのでしょうか？　それを考えるために，2つのイースをよく見てみることにしましょう。まず，やった人ならわかるでしょうが，セガ版は当たり判定がかなり甘く，敵からのダメージを受けないように半分ずらしての攻撃は楽なのですが，ボスキャラを倒すときなどはそれが裏目に出てとても倒しにくいです。このことから，イースの売り物のひとつであるゲームのやさしさは消えてしまったと言わざるを得ません。

だから，私はイースにのめり込めるのは，ゲームがやさしいからというわけではないと思います。また，私のセガマークⅢは，FM音源ボードを未だに付けていないために，ゲーム中のBGMもPSGでしか聞けないのですが，編集室で聞いたX1版のPSGのみのBGMに比べてもかなり違う感じがします。どうもLFOがかかりすぎてるみたいです。

なんか，こんなことを言ったらセガ版を作った人には申しわけないですけど，BGMが「へろへろへろー」としてしまって，お世辞にもノリのいい音楽とは言えそうにもありません。つまり音楽のノリでゲームにのめってるわけでもないのです。

それでも，どちらのイースもやはり楽しいんですよね。どちらのイースにも共通しているもの，それはストーリーと演出です。私が思うに，やはりイースは，ゲームのやさしさでも音楽でもなく，「いまここで戦えば，また新しい世界が見られるんだよ」という，演出のうまさだと思います。

これは，実際の私たちの生活にはあり得ない，戦闘という行為が終わったあとこれまでと違う自分がいる。心のどこかにある，「いまよりも違う自分でありたい」という願望が叶えられるに違いないと思わせる。そんな感じだと思うんですよね。

ジャンプ好きのひとり言

さあて，話はコロッと変わってしまいまして（なに，そんなにコロコロ変えるなって？　しょうがないでしょ，私はB型なんだ，血液型が)。いきなり少年ジャンプの話です。

最近，私は久しぶりに聖闘士星矢の単行本を買いました。うーん，あれって，ほーんとに戦ってばっかりですねー。単行本で見てみると「アテナの胸に突き刺さった矢をアテナの楯で消し去るために星矢，紫龍，瞬，氷河（たまーに一輝も出てくるけど）の4人の聖闘士が12の宮を突破する」というストーリーのためだけに，単行本何冊にもわたって戦いっぱなし。ほとんど，ワードナを倒したいがために，迷宮のなかで核攻撃の呪文「TILTOWAIT！」を覚えるまで，黙々と敵と戦ってしまう，ウィザードリィのキャラのようです（あ，RPGなら基本的にはなんでもそうか)。

ドラゴンボールにしろ，バスタードにしろ，星矢ほどではないにせよ似たような感じで突き詰めてあらすじを言ってしまうと，すべて，「敵の大ボスを倒すためにとりあえず目の前の敵と戦うマンガ」に到達すると言えそうな気もします。最近のメジャーなマンガっていうと，だいたい少年ジャンプ，ヤングジャンプ，ビッグコミックスピリッ

ツといったメンバーだと思いますが，私はヤンジャン（ヤングジャンプのことね）はあまり読んでいないのでわからないけども，たぶん一般的な傾向としてメジャーな，というより売れているマンガっていうのには戦うとか，勝負に勝つっていうマンガが多いんですよね，昔から。

逆に実際の普通の生活っていうと「朝起きて，昼学校にいって，夜寝た」てなもんなはずですが，そんなのんべんだらりとしたストーリーの生活マンガなんて見たことないですよね（見たくないって，んなもん）。

実際の生活のうえでは，勝負なんていうと所属している部活の試合とか受験戦争ぐらいしかないのに，なぜかまったく知らない世界に首を突っ込もうとする。不思議じゃありませんか，こんなのって。

この疑問を解決するために，私は学校で買ったのになぜかほとんど開いたことのない，心理学の授業の教科書に載っていた用語の「投影」という言葉を思い出しました。

「投影(映)，これは自己にとってコントロールできない性質をほかの人やものに転嫁して緊張を解消するメカニズムである」

なるほど，マンガのキャラが実は読んでる人の理想像なのか。でもって，ああいう勝負がしてみたいわけね。

よーく，思い出してごらんよ

ここで，ゲームだけでなく「いままで自分がよく見たもので，かつ周りの人間もよく見たもので話題にできたような話」，というものについて考えてみたいと思います。ちょっと，条件が長ったらしいですか？もっと具体的に言うと，幼いころに見た「仮面ライダー」であったり，ジャンプで読んでいる「シティーハンター」でもいいし，予約して発売日に買ったX68000の「琥珀色の遺言」でも構いません。よーするに，マンガでもテレビでも，ゲームでも自分が好きなもので，なおかつはやったものならなんでもいいのです。誰もが仮面ライダーの勇姿に憧れ，たまーに出てくる僚（冴羽僚です。シティーハンターの）のハードボイルドな姿に酔い，または大正時代に生きる人たちとともに探偵になった自分は，ひとりの男の生きざまなどを見て酔っていたはずなのです。

感動といえばですね，幼いころウルトラマンや仮面ライダーを見るためにテレビの前にかじり付き，仮面ライダーがキックをするときにライダーと一緒に「ライダーキック！」なんて叫んだり，「ライダー危機一髪」なんてときにはライダーが死んじゃったと泣いたことはありませんか？　それで，番組を見終わると買ってもらった仮面ライダーベルト（確か，電池が入っててスイッチ入れるとなかで羽根がクルクル回って音も出るという。幼稚園の年少さんのときだったかなー，私も持ってたんですよねー，アレって）を付けて，補助輪の付いた自転車で出て行って，友だちとライダーごっこをしに行ったとか。なぜか，私はそういうのをやらせると怪獣の役ばかりでしたけど。ゴレンジャーのときはキレンジャーばっかだったし（ブツブツ）。

ま，それはともかく，あのころってもう純粋に感動しちゃって，仮面ライダーが存在すると信じて疑わないんですよねー。私はあれこそが，感動するということのいちばん正しい姿なのではないかと思います。

いま，どんなにリアリティのあるストーリーでゲームを作ったとしても，かつて仮面ライダーを見て「カッコいいなあ」と思ったときの感動には勝つことはできないでしょう。

自分ともうひとりの自分。物語のキャラに憧れる，あるいはいつも自分が体験しないもの，できないものをマンガ，小説，テレビなどの物語で体験して楽しんで，私たちは育ってきたはずなのです。その数多くの経験こそがいまの感動の源となっているのです。

そして，究極のゲームへ

さて，話はまたゲームのほうへ戻りますけど，イースIIのオープニングのアニメーションがありますよね。私はあれの某AV版をデジタルディスプレイで見ていてわかったのですが，あれって面白いことをしているんですね。画面に表示されていない，余ったVRAMに一度（たぶん字幕を表示しているときにのんびりと）絵を書き込んでおいて，そのあと一気に表示されているVRAMにデータを転送しているんですねー。うーん，面白い。

6809での話だけど，どうしても高速にメモリ上のデータのブロック転送をしたいときにはPUSH，PULL命令を使うとか，命令を展開しておいてループを組まずに実行するとかの基本的な手法は，昔っから使われてきたんですよね。少なくとも，その方法はPC-88でmarkIIが全盛のころに，高速化CLSのアルゴリズムとして雑誌に載ったのを私は覚えているし，たぶんこれはMZでも使われていたと思います。

そんな当たり前の技術で「あー，すごいなー」と思わせるっていうのは，たいしたもんですよね。技術に頼らずに（別に頼ってもいいけど），映画などを意識して見せ方を工夫して大作ゲームを作るようになったこの時代の代表みたいなもんですよね，あのオープニングは。

ところで前のほうで，あっさりと受験という戦いぐらいしかないはずだと私は言いましたけど，考えてみれば高校受験にせよ大学受験にせよ，受験というのは恐ろしいほどゲーム的な要素を持っているわけです。たとえば，本人の裁量に多くを任せられた「合格」という最終目標があって，で，同じ学校の同じ学科を受けるライバルという多くの敵キャラがいて，試験問題という障害があって，それらを効率的にクリアすること。これらすべてが人為的ルールの上で動いているんですよねー，困ったことに。

ところが，どっこい。マンガにせよ，ビデオにせよ，そして，当然のようにパソコンゲームにもそういう題材はほとんど取り上げられることはなかったのです。当然ですよね。そういうゲームは，どんなにいい見せ方をしようとしても，そのゲームという名の夢から一気に現実に引き戻すようなテーマがその夢をぶっ壊してしまうからなのです。あの，イースIIも前作の世界がしっかりとあったからこそ，ああいう見せ方が成功したのです。

本当にいいゲームを作ろうと思ったら，夢を見せるという目的とそれを見せる方法の両方がしっかりしてなくては，本当にいいゲームとは言えないのです。

やはりゲームというのは，人にとって願望を実現してくれる言わば心の鏡のような存在であってほしいものです。これからのゲームには，やはり心の底の願望をイースでザコキャラを倒すがごとく，プレイ中にいつでも満たしてくれるものであってほしいものですよねー。

Hyper Game Book

Mounai Toshiyuki
毛内 俊行

あのHyper Cardの基本機能さえ削除したといわれる伝説のゲームブック作成ツールがついに登場。その名もHyper Game Bookです。MZ-2500のBASICを利用した簡単なプログラムですが，シナリオしだいで本格的なアドベンチャーが楽しめます。

皆さん，長い間お待たせしました。えっ，冗談だと思っていたって？　実は本当に作っていたんですよ。なにをって，そりゃあもちろんMZ-2500用Hyper Game Bookのことだってば！

いったいなにができるのか?

Hyper Game Bookでなにができるのか。このプログラムはゲームブックというように，文庫本でいっぱい出ている「サイコロ本」を忠実にシミュレートしたものなんです。

基本的な機能は1988年6月号で予告(？)したものと同じで，簡易言語風のシナリオデータをコンバートして実行するものです。シナリオデータはBASICから呼び出せるアルゴエディタを使って入力するように設計されていますが，アルゴエディタを持っていない人でもデータを作成することは可能です。その方法については49ページの囲み記事で詳しく説明しましょう。

それでは，MZ-2500のBASIC-M25を起動してください。メモリの拡張などは一切必要ありません。プログラムはすべてBASICで書かれているので，気軽にキーボードをポコポコ叩いて入力するだけです。

シナリオデータの作成

Hyper Game Bookは，本体だけではなにもできません。このプログラムはシステムに相当し，遊ぶためには肝心なゲームシナリオを作成しなければならないからです。ここでは，コンバートする前の編集可能なファイルをシナリオデータ，コンバートされたものをゲームファイルと呼ぶことにしましょう。

それでは，実際にシナリオデータを入力する場合の，データの文法について説明しましょう。といっても文法などというほどのものではなく，わずか5つの命令をマスターすれば，誰でも簡単にシナリオが書けるようになっています。5つの命令の使い方は以下のとおりです。基本的に表示したいメッセージデータ以外は，必ず半角文字にしてください。エラー発生の原因になります。特に「}」や「,」には気をつけましょう。なお，文例中に出てくる「n」は1以上の整数，「data」は文字列を示しています。

```
*n
シーン番号の設定
```

データのシーン番号をセットします。たとえば，シーン1をセットするときは，

　＊1

とします。

シナリオデータの各シーンは，必ずこの命令から始まらなくてはなりません。コンバータは，この命令を見つけることによって，次のシーンに進んだことを理解するからです。ちょうど，本でいうページ番号にあたるものです。また，シーン番号は，1から順番に大きくなっていかなくてはなりません。つまり，一番最初のシーン番号は必ず1，その次は2ということになります。あと，細かいことですが，番号は必ず「＊」の直後に置いてください。スペースを開けると読めなくなります。

```
M{,data,}
メッセージ表示
```

メッセージ，つまり本文を表示します。dataは何行にわたって書いても問題ありません。データ中に「}」を見つけた時点でデータの終了を示します。また，

　M{,どうしますか？,/,}

のように，データ中に「/」を単独に用いた場合は改行コードの役割となります。

なお，ほかの命令についてもいえることですが，命令とデータ，または命令と命令，データとデータなどの間は，必ず「,」か改行コードで区切り，「"」のようなBASICのINPUT文では直接入力できないキャラクタは使わないでください。

```
W{,data1,data2,…,}
メニューワードのセット
```

これは早い話が選択文のセットです。データは半角文字で16文字，漢字などの全角文字で8文字まで使えます。また，選択文は一度に4つまで指定できます。この命令で設定した選択文は，もう一度この命令を実行しないかぎり，シーンが変わっても有効です。

```
G{,n1,n2,…,}
条件つきジャンプ
```

メニューの選択に対応したジャンプを行います。これは先ほどのW命令に対応しており，たとえばシナリオ中に，

　W{,YES,NO,}
　G{,12,35,}

というデータがあったなら，ゲーム実行中にメニューから，YESを選択すればシーン番号35へジャンプしろという意味になります。つまりこの命令も，指定できるシーン番号は最大4つまでです。

```
J{,n,}
無条件ジャンプ
```

この命令が実行されると，有無をいわさず指定されたシーンへジャンプします。選択肢がない場合に用います。

```
END
実行終了
```

この命令が実行されると，データの実行を終了します。要するにゲームオーバーです。

さあ，以上の５つの命令をマスターするだけで，シナリオデータが作成できるのです。

いよいよ実行

Hyper Game Bookのシステムをドライブ1，ゲームファイルを作るためのデータディスクをドライブ2にセットしておきます。データをコンバートするときは，一応シナリオデータの入ったディスクをドライブ1に入れ，ゲームファイルをドライブ2に生成するようになっています。以下，2ドライブ対応の場合について説明します。ただ，プログラムを1カ所書き換えればディスクドライブが1基しかない方でも使用できるようになります。この場合の変更点についても49ページの囲み記事をご覧ください。

データディスクはBASICのスレーブディスクでかまいませんが，コンバートされたゲームファイルはデータのサイズが意外と大きくなるので，新しいディスクを用意するとよいでしょう。

Hyper Game Bookのシステムを起動すると，画面に2つのメニューが表示されます。カーソルのある小さいメニューが機能メニュー。その下のなにも書かれていない広いメニューが，シナリオ選択メニューで，シナリオファイルをコンバートすると，ここにシナリオ名が表示されます。

機能メニューでは，以下の4つの機能が選択できます。カーソルキーで目的の機能を選択し，リターンキーで決定してください。

- **ゲーム**

  コンバートしたゲームを実行します。

- **作成**

  アルゴエディタで作成したファイルのコンバートを行います。

- **削除**

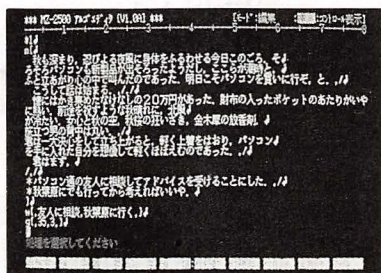
  作成したゲームファイルの削除を行います。

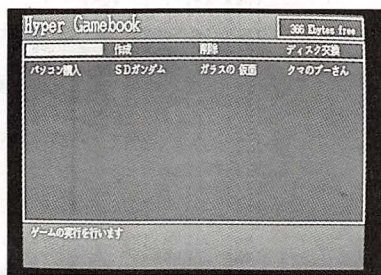
- **ディスクの交換**

  データディスクの交換を行います。

以上の説明は，メニュー画面の下に常時表示されるので迷うことはないでしょう。また，選択を取り消したいときはESCキーでキャンセルできます。

Hyper Game Bookの使い方

まずはアルゴエディタを使ってシナリオデータを入力しましょう。

シナリオデータ用のディスクにセーブします。Hyper Game Bookのシステムといっしょにセーブしておいてもかまいません。

Hyper Game Bookのシステムを起動します。

ドライブ1にシナリオデータの入ったディスクを入れ，ドライブ2にゲーム用ディスクを入れて作成を選択。コンバートしたいシナリオデータのファイル名を入力してコンバート開始。待つこと数分。

ゲームの実行。プレイしたいゲームファイルを選択しましょう。

オープニングタイトルが表示されて

ゲーム画面はこのとおり

シナリオを変更したい場合や，コンバート中にエラーが出たりした場合はすかさずアルゴキーを押してアルゴエディタでシナリオデータの再編集が可能。さすがはアルゴ機能。

リスト1 Hyper Game Book

```
1000 '
1010 'H y p e r   G a m e b o o k
1020 '
1030 'version 1.0c : 1988/09/06 12:23
1040 '
1050 '製作：毛内俊行
1060 '
1070 '
1080 *START
1090 INIT "crt1:,,,0"
1100 INIT "crt2:320,200,16"
1110 KLIST 0
1120 CLS 3
1130 LINE (0,0)-(319,199),6,BF
1140 LINE (0,21)-(319,35),15,B
1150 LINE (1,22)-(318,34),0,BF
1160 LINE (2,23)-(318,34),4,BF
1170 LINE (0,35)-(319,163),15,B
1180 LINE (1,36)-(318,162),0,BF
1190 LINE (2,37)-(318,162),4,BF
1200 LINE (0,164)-(319,164),0
1210 LINE (240,2)-(319,18),15,B
1220 LINE (241,3)-(318,17),0,BF
1230 LINE (242,4)-(318,17),4,BF
1240 LINE (240,19)-(319,19),0
1250 SYMBOL (3,1),"Hyper Gamebook",,,0
1260 SYMBOL (2,0),"Hyper Gamebook",,,15
1270 '
1280 *START2
1290 CONSOLE 21,4
1300 CLS
1310 CONSOLE 21,3
1320 CLEAR
1330 STOP ON
```

エラーメッセージ

コンバート中に発生するエラーメッセージは以下のとおりです。

“文法の誤りです”

ご存じ，SYNTAX ERRORというやつです。指定されたシーンの前後にシナリオデータの記述ミスが発見できるはずです。

“シーン番号0の指定はできません”

シーン番号は1から指定可能です。いきなりシナリオの先頭に，＊0なんて書いてませんか？

“シーン番号が連続していません”

シーン番号はひとつずつ増えていかなくてはなりません。

“ジャンプ先の指定は4つまでです”

条件分岐命令で，4つ以上の分岐をしようとしたときに発生しますが，G命令の終わりの｝を忘れたり，全角文字で書いたりしてもエラーとなります。

“選択文は1シーン4つまでです”

G命令のエラーと同じです。

“ジャンプ先がありません”

ジャンプ先のシーン番号が見つからないときに発生するエラーです。

“シナリオ管理ファイルがいっぱいです”

ゲームディスクに60個以上のゲームファイルを作ろうとすると発生するエラーですが，通常はその前にディスクの容量がいっぱいになってしまうでしょう。いずれにしても，このエラーを見た人はすごい人です。

“すでに同名のファイルが存在します”

同じ名前のゲームファイルは登録できません。名前を変えるか，古いファイルを削除してください。

“ジャンプ先に0を指定しています”

シーン番号に0が指定できないのだから，ジャンプ先も0はありえません。

“エラーいこっちゃ！　その他のエラーだ”

しまったぁ！　こいつは予想もしなかったぞ。というようなエラーです。前述のすべてのエラーチェックに引っ掛からず，BASICがエラーを起こしてしまったときに発生します。

＊　＊　＊

このプログラムはオールBASICではありますが，「たかがBASIC」とはいえないくらいMZ-2500のBASICの実力を発揮したプ

```
1340 DRV1$="1:" ' ソースファイルの入っているドライブ名
1350 DRV2$="2:" ' シナリオの作成，管理をするドライブ名
1360 LOCATE 62,1:PRINT DEVF(DRV2$);"Kbytes free "
1370 OPEN "r",#1,DRV2$+"シナリオ管理",32
1380 FIELD #1,16 AS A$
1390 PUT #1,1
1400 GET #1,1
1410 CHECK=EOF(#1)
1420 CLOSE
1430 IF CHECK=-1 THEN *NEWFORMAT
1440 '
1450 DIM JPT(4),KYT$(4),MES$(15)
1460 ON STOP GOSUB *QUIT
1470 ON ERROR GOTO *ERROR
1480 DIRNB=0
1490 FILEMAX=61
1500 CONSOLE 0,24,0,80
1510 LOCATE 0,3
1520 PRINT TAB(2);"ゲーム";
1530 PRINT TAB(22);"作成";
1540 PRINT TAB(42);"削除";
1550 PRINT TAB(62);"ディスク交換"+SPACE$(5)
1560 PRINT
1570 '
1580 OPEN "r",#1,DRV2$+"シナリオ管理",32
1590 FIELD #1,16 AS FLN$
1600 FOR I=2 TO 61
1610   GET #1,I
1620   IF FLN$<>SPACE$(16) THEN
1630     PT$=SPACE$(2)+FLN$+SPACE$(18)
1640     PRINT LEFT$(PT$,20);
1650   END IF
1660 PRINT CHR$(5);
1670 NEXT I
1680 PRINT
1690 CLOSE
1700 X=2:Y=3:CMC=1
1710 GOSUB *CUR
1720 IF DRNM$="ゲーム" THEN CM=1
1730 IF DRNM$="作成" THEN CM=2
1740 IF DRNM$="削除" THEN CM=3
1750 IF DRNM$="ディスク交喚" THEN CM=4
1760 X=2:Y=5:CMC=0
1770 IF CM=1 OR CM=3 THEN GOSUB *CUR
1780 ON CM GOTO *GAME,*MAKE,*KILL,*EXC
1790 '
1800 *CUR
1810 KEY 0,""
1820 CONSOLE 21,3,2,76
1830 CLS
1840 CREV@ (X-1,Y)-(X+16,Y),1
1850 REPEAT
1860   IF CMC=1 THEN GOSUB *COMMS
1870   KY=ASC(INKEY$)
1880   IF KY<>0 THEN
1890     CLS
1900     CREV@ (X-1,Y)-(X+16,Y),0
1910     IF KY=&H1C AND X<60 THEN X=X+20
1920     IF KY=&H1D AND X>2 THEN X=X-20
1930     IF KY=&H1E AND Y>5 AND CMC=0 THEN Y=Y-1
1940     IF KY=&H1F AND Y<19 AND CMC=0 THEN Y=Y+1
1950     IF KY=&H1B AND CMC<2 THEN *START2
1960     CREV@ (X-1,Y)-(X+16,Y),1
1970   END IF
1980 UNTIL KY=13
1990 DRNM$=SCRN$(X,Y,16)
2000 FOR I=16 TO 1 STEP -1
2010   IF MID$(DRNM$,I,1)<>" " THEN 2030
2020 NEXT I
2030 DRNM$=LEFT$(DRNM$,I)
2040 IF DRNM$="" THEN 1840
2050 RETURN
2060 '
2070 *QUIT
2080 STOP OFF
2090 CLOSE
2100 INIT "crt1:,,,0
2110 CLS 3
2120 END
2130 '
2140 *COMMS
2150 LOCATE 2,21
2160 IF X=2 THEN
2170   PRINT "ゲームの実行を行います"
2180 ELSE IF X=22 THEN
2190   PRINT "アルゴエディタで作成したファイルをコンバートしま
す"
2200 ELSE IF X=42 THEN
2210   PRINT "作成したファイルの削除します"
2220 ELSE IF X=62 THEN
2230   PRINT "データーディスクの交喚を行います"
2240 END IF
2250 RETURN
2260 '
2270 *DIRMAKE
2280 IF DIRNB*FILEMAX>SCENE THEN RETURN
2290 DIRNB=DIRNB+1
2300 DIRNAM$=DRNM$+STR$(DIRNB)
2310 MKDIR DRV2$+DIRNAM$
2320 RETURN
2330 '
2340 *ERKILL
2350 ERC=1
2360 GOTO 2540
2370 *KILL
2380 ERC=0
```

ログラムになっているはずです。ゲーム中のページスクロールなどは感動ものです。ただし，ファイルのコンバート速度が遅いのが気になるかもしれません。これはファイル構造の都合でBASICのせいではありません。大きなシナリオだと４～５分かかることもあるので，コーヒーでも飲みながら気長に待ちましょう。

もうこれ以上説明することはありません。あとは皆さんがこのプログラムをどう活用するかです。友達同士でシナリオをたくさん作って遊ぶのもよいでしょうし，プログラムを解析して新たなシステムを作るのもよいでしょう。このシステムにいろいろなパラメータ管理や乱数を加えるともっと面白くなると思います。皆さんなりの活用を期待します。

アルゴエディタのない方へ

シナリオデータは，V2用のBASIC についているアルゴエディタから入力すると便利なように作られていますが，旧バージョンの BASIC しか持っていない方もいるでしょう。アルゴエディタを使わずにデータを作成するには次のようにBASICのエディタを使用します。BASICから，

auto＊

と実行して，BASIC の REM 文の形式でデータを入力します。データの書式はアルゴエディタを使う場合と同様です。

完成したら，

SAVE“ファイル名１”

としてセーブしておきましょう。しかし，このままではコンバートできませんので，

LIST＊“ファイル名２”

としてください。すると，あら不思議。ゲームシナリオにコンバート可能なシナリオデータがファイル名２というかたちでディスクにセーブされるようになっています。なお，再びシナリオをエディットするにはファイル名１のファイルが必要なので消さずに残しておきましょう。

１ドライブで使用する場合

ディスクドライブが１基しかない場合には，２つの方法があります。ひとつは単純に同じドライブを使う方法で，ドライブ１→ドライブ１でゲームファイルを作成します。1350行のドライブ指定を書き換えてください。

1350行　DRV2$＝“2:”　→　DRV1$＝“1:”

この場合，同じディスクにもとのシナリオデータとゲームファイルが共存するため容量が多少小さくなってしまいます。多少面倒でも目一杯１枚分のゲームファイルを作りたいという場合には，データレコーダを使うという手もあります。シナリオデータをテープから読み込み，ドライブ１にゲームファイルを作成するわけです。この場合，1350行の変更に加えて，1340行も次のように変更してください。

1340行　DRV1$＝“1:”→ DRV1$＝“CMT:”

```
2390 COLOR 2
2400 BEEP
2410 PRINT "削除してもいいですか？ yes=[CR]"
2420 COLOR 7
2430 A$=INKEY$
2440 IF A$="" THEN 2430
2450 IF A$<>CHR$(13) THEN
2460   PRINT "中止します．"
2470   GOTO *START2
2480 ELSE
2490   PRINT "削除します．"
2500 END IF
2510 '
2520 CONSOLE 21,4
2530 LOCATE 2,22
2540 OPEN "i",#1,DRV2$+DRNM$+" 1/ファイル数"
2550 INPUT #1,A$
2560 CLOSE
2570 DIRMAX=0
2580 ERSC=VAL(A$)
2590 IF ER=6 THEN ERSC=SCENE
2600 IF A$="0" THEN
2610   KILL DRV2$+DRNM$+" 1/ファイル数"
2620   RMDIR DRV2$+DRNM$+" 1"
2630   GOTO 2760
2640 ELSE
2650   FS=VAL(A$)
2660   FOR I=1 TO FS
2670     GOSUB *ASKDIR
2680     IF ERC=0 THEN PRINT:PRINT "Now deleting - ";SCFL$;
2690     KILL DRV2$+SCFL$
2700   NEXT I
2710   KILL DRV2$+DRNM$+" 1/ファイル数"
2720   FOR I=1 TO DIRMAX
2730     RMDIR DRV2$+DRNM$+STR$(I)
2740   NEXT I
2750 END IF
2760 OPEN "r",#1,DRV2$+"シナリオ管理",32
2770 FIELD #1,16 AS FLN$
2780 FOR I=2 TO 61
2790   GET #1,I
2800   IF FLN$=LEFT$(DRNM$+SPACE$(16),16) THEN
2810     LSET FLN$=SPACE$(16)
2820     PUT #1,I
2830     GOTO 2860
2840   END IF
2850 NEXT
2860 CLOSE
2870 IF ERC=0 THEN *START2
2880 IF ERSC<>0 THEN
2890 'cls
2900   COLOR 2
2910   PRINT "ただいまのエラーは scene";ERSC;"で発生しました"
2920   COLOR 7
2930   BEEP
2940 END IF
2950 PRINT "push [CR] key";
2960 IF INKEY$<>CHR$(13) THEN 2960
2970 GOTO *START2
2980 '
2990 *ASKDIR
3000 DRN=INT((I-1)/FILEMAX)+1
3010 IF DRN>DIRMAX THEN DIRMAX=DRN
3020 SCFL$=DRNM$+STR$(DRN)+"/scene"+STR$(I)
3030 RETURN
3040 '
3050 *FORMAT
3060 BEEP
3070 COLOR 2
3080 PRINT "シナリオ管理ファイルの初期化を行います．よろしいで
すか？ yes=[CR]"
3090 COLOR 7
3100 A$=INKEY$
3110 IF A$="" THEN 3100
3120 IF A$<>CHR$(13) THEN
3130   PRINT "中止します．"
3140   GOTO *START2
3150 END IF
3160 '
3170 *FORMAT2
3180 OPEN "r",#1,DRV2$+"シナリオ管理",32
3190 FIELD #1,16 AS FLN$
3200 FOR I=2 TO 61
3210   LSET FLN$=SPACE$(16)
3220   PUT #1,I
3230 NEXT
3240 CLOSE
3250 CONSOLE 5,20
3260 CLS
3270 GOTO *START2
3280 '
3290 *NEWFORMAT
3300 CONSOLE 21,3,2,76
3310 COLOR 2
3320 PRINT "シナリオ管理ファイルがありません．新しく作成します"
3330 COLOR 7
3340 GOTO *FORMAT2
3350 '
3360 *EXC
3370 LOCATE 2,21
3380 PRINT "新しいデーターディスクをセットして[CR]キーを押して
下さい"
3390 BEEP
3400 IF INKEY$<>CHR$(13) THEN 3400
3410 GOTO 3250
3420 '
```

```
3430 *MAKE
3440 INPUT "ファイル名=",DATNAM$
3450 DATNAM$=LEFT$(DATNAM$,14)
3460 COLOR 2
3470 PRINT CHR$(&H1E)+"ファイル名="+DATNAM$+CHR$(5)+CHR$(7)
3480 COLOR 7
3490 PRINT "作成してよろしいですか? yes=[CR]"
3500 REPEAT
3510 A$=INKEY$
3520 UNTIL A$<>""
3530 IF A$<>CHR$(13) THEN
3540   PRINT "中止します."
3550   GOTO *START2
3560 END IF
3570 '
3580 CONSOLE 21,4
3590 LOCATE 2,22
3600 OPEN "r",#1,DRV2$+"シナリオ管理",32
3610 FIELD #1,16 AS FLN$
3620 FOR I=2 TO 61
3630   GET #1,I
3640   IF DATNAM$=LEFT$(FLN$,LEN(DATNAM$)) THEN 3750
3650   IF FLN$=SPACE$(16) THEN 3790
3660 NEXT I
3670 CLOSE
3680 ER=7
3690 CLOSE
3700 GOSUB *ERRPRT
3710 PRINT "push [CR] key"
3720 REPEAT
3730 UNTIL INKEY$=CHR$(13)
3740 GOTO *START2
3750 '
3760 ER=8
3770 GOTO 3690
3780 '
3790 DRNM$=DATNAM$
3800 SCENE$=""
3810 SCENE=0
3820 BACKSC=0
3830 '
3840 CLOSE
3850 OPEN "i",#1,DRV1$+DATNAM$
3860 '
3870 *MAIN
3880 DLINE=0
3890 IF EOF(#1) THEN
3900   SC=SCENE
3910   GOSUB *SCSET
3920   GOTO *JPCHK
3930 END IF
3940 '
3950 INPUT #1,DAT$
3960 IF LEFT$(DAT$,1)="*" THEN *SCENESET
3970 IF DAT$="w{" OR DAT$="W{" THEN *KEYSET
3980 IF DAT$="m{" OR DAT$="M{" THEN *MESSET
3990 IF DAT$="g{" OR DAT$="G{" THEN *GOTSET
4000 IF DAT$="j{" OR DAT$="J{" THEN *JMPSET
4010 IF DAT$="end" OR DAT$="END" THEN *ENDSET
4020 IF DAT$="" THEN *MAIN
4030 '
4040 CLOSE
4050 ER=1
4060 GOSUB *ERRPRT
4070 SC=SCENE
4080 GOSUB *SCSET
4090 DRNM$=DATNAM$
4100 GOTO *ERR
4110 '
4120 *ERRPRT
4130 BEEP
4140 PRINT
4150 IF ER=0 THEN
4160   COLOR 2
4170   PRINT "エラーいこっちゃ!  その他のエラーだ."
4180   COLOR 7
4190 ELSE IF ER=1 THEN
4200   PRINT "文法の誤りです."
4210 ELSE IF ER=2 THEN
4220   PRINT "シーン番号0の指定はできません."
4230 ELSE IF ER=3 THEN
4240   PRINT "シーン番号が連続していません."
4250 ELSE IF ER=4 THEN
4260   PRINT "ジャンプ先の指定は4つまでです."
4270 ELSE IF ER=5 THEN
4280   PRINT "選択文は1シーン4つまでです."
4290 ELSE IF ER=6 THEN
4300   PRINT "ジャンプ先がありません."
4310 ELSE IF ER=7 THEN
4320   PRINT "シナリオ管理ファイルがいっぱいです."
4330 ELSE IF ER=8 THEN
4340   PRINT "すでに同名のファイルが存在します."
4350 ELSE IF ER=9 THEN
4360   PRINT "ジャンプ先に0を指定しています."
4370 END IF
4380 RETURN
4390 '
4400 *SCSET
4410 CLOSE
4420 OPEN "o",#1,DRV2$+DRNM$+" 1/ファイル数"
4430 PRINT #1,SC
4440 CLOSE
4450 GOSUB *FNSET
4460 RETURN
4470 '
4480 *JPCHK
4490 PRINT
4500 PRINT "ジャンプテーブルのチェックを行います.もう少しお待ち下さい.";
4510 DIRNB=0
4520 FOR SCENE=1 TO SC
4530   PRINT:PRINT "Now check - scene";SCENE;
4540   DIRNB=INT(SCENE/(FILEMAX+1))+1
4550   DIRNAM$=DRNM$+STR$(DIRNB)
4560   OPEN "r",#1,DRV2$+DIRNAM$+"/scene"+STR$(SCENE),80
4570   FIELD #1,76 AS FLN$
4580   FOR I=1 TO 4
4590     GET #1,I
4600     IF SC<VAL(FLN$) THEN
4610       CLOSE
4620       ER=6
4630       GOSUB *ERRPRT
4640       DRNM$=DATNAM$
4650       GOTO *ERR
4660     END IF
4670   NEXT I
4680 CLOSE
4690 NEXT SCENE
4700 GOTO *START2
4710 '
4720 *FNSET
4730 CLOSE
4740 OPEN "r",#1,DRV2$+"シナリオ管理",32
4750 FIELD #1,16 AS FLN$
4760 FOR I=2 TO 61
4770   GET #1,I
4780   IF FLN$=SPACE$(16) THEN 4800
4790 NEXT
4800 LSET FLN$=DATNAM$
4810 PUT #1,I
4820 CLOSE
4830 RETURN
4840 '
4850 *ERR
4860 PRINT "エラー後の処理を行っています.しばらくお待ち下さい"

4870 GOTO *ERKILL
4880 '
4890 *SCENESET
4900 GOSUB *DIRMAKE
4910 SCENE=VAL(MID$(DAT$,2))
4920 IF SCENE=0 THEN
4930   ER=2
4940   GOTO 4980
4950 END IF
4960 IF SCENE<>BACKSC+1 THEN
4970   ER=3
4980   GOSUB *ERRPRT
4990   SC=BACKSC
5000   GOSUB *SCSET
5010   DRNM$=DATNAM$
5020   GOTO *ERR
5030 END IF
5040 BACKSC=SCENE
5050 CLOSE #2
5060 OPEN "r",#2,DRV2$+DIRNAM$+"/scene"+STR$(SCENE),80
5070 FIELD #2,76 AS SDAT$
5080 PRINT:PRINT "Now making scene";STR$(SCENE);
5090 GOTO *MAIN
5100 '
5110 *JMPSET
5120 INPUT #1,A$
5130 IF VAL(A$)=0 THEN
5140   ER=9
5150   GOTO 5380
5160 END IF
5170 LSET SDAT$=A$
5180 PUT #2,1
5190 LSET SDAT$="*"
5200 PUT #2,9
5210 LSET SDAT$=""
5220 FOR I=2 TO 8
5230   PUT #2,I
5240 NEXT
5250 INPUT #1,A$
5260 IF A$="}" THEN *MAIN
5270 ER=1
5280 GOTO 5380
5290 '
5300 *KEYSET
5310 FOR I=5 TO 9
5320   INPUT #1,A$
5330   IF A$="}" THEN *MAIN
5340   LSET SDAT$=A$
5350   PUT #2,I
5360 NEXT
5370 ER=4
5380 GOSUB *ERRPRT
5390 SC=SCENE
5400 GOSUB *SCSET
5410 DRNM$=DATNAM$
5420 GOTO *ERR
5430 '
5440 *GOTSET
5450 FOR I=1 TO 4
5460   INPUT #1,A$
5470   IF A$="}" THEN *MAIN
5480   IF VAL(A$)=0 THEN
5490     ER=9
5500     GOTO 5380
5510   END IF
5520   LSET SDAT$=A$
```

▶知人の家でX68000の源平をプレイさせてもらい，年末にX68000を買いたいと思って，初めてOh!Xを買いました。絶対にX68000を買うぞー。いまからお金を貯めています。もし買えたらまた連絡したいと思います。では。　　金子　和祐　(19)　奈良県

```
5530   PUT #2,I
5540 NEXT
5550 ER=5
5560 GOTO 5380
5570 '
5580 *ENDSET
5590 FOR I=1 TO 4
5600   LSET SDAT$="0"
5610   PUT #2,I
5620 NEXT
5630 LSET SDAT$="終了"
5640 PUT #2,5
5650 FOR I=6 TO 9
5660   LSET SDAT$=SPACE$(16)
5670   PUT #2,I
5680 NEXT
5690 GET #2,9
5700 IF EOF(#2) THEN
5710   LSET SDAT$=""
5720   PUT #2,10
5730 END IF
5740 GOTO *MAIN
5750 '
5760 *MESSET
5770 C$=""
5780 *MESSET2
5790 INPUT #1,A$
5800 IF A$="}" THEN
5810   LSET SDAT$=C$+SPACE$(76)
5820   PUT #2,DLINE+10:DLINE=DLINE+1
5830   GOTO *MAIN
5840 END IF
5850 IF KACNV$(A$)="/" THEN
5860   LSET SDAT$=C$+SPACE$(76)
5870   PUT #2,DLINE+10:DLINE=DLINE+1
5880   GOTO *MESSET
5890 END IF
5900 C$=C$+A$
5910 IF LEN(C$)<76 THEN *MESSET2
5920 '
5930 Z=ASC(MID$(C$,76,1))
5940 IF (Z>&H80 AND Z<&HA0) OR (Z>&HDF AND Z<&HFD) THEN
5950   Z=ASC(MID$(C$,75,1))
5960   IF (Z>&H80 AND Z<&HA0) OR (Z>&HDF AND Z<&HFD) THEN
5970     Z=76
5980   ELSE
5990     Z=75
6000   END IF
6010 ELSE
6020   Z=76
6030 END IF
6040 '
6050 LSET SDAT$=LEFT$(C$,Z)
6060 PUT #2,DLINE+10:DLINE=DLINE+1
6070 C$=MID$(C$,Z+1)
6080 GOTO *MESSET2
6090 '
6100 *GAME
6110 DATNAM$=DRNM$
6120 DRNM$=DATNAM$
6130 CONSOLE 0,24,0,80
6140 CLS 1
6150 LINE (0,21)-(319,148),15,B
6160 LINE (1,22)-(318,147),0,BF
6170 LINE (2,23)-(318,147),4,BF
6180 LINE (0,148)-(319,163),15,B
6190 LINE (1,149)-(318,162),0,BF
6200 LINE (2,150)-(318,162),4,BF
6210 LOCATE 0,7
6220 PRINT TAB(28);"Hyper Gamebook ver1.0":PRINT
6230 PRINT TAB(25);"Programed by Toshiyuki Mounai ":PRINT
6240 PRINT TAB(39-LEN(DATNAM$)/2);DATNAM$:PRINT
6250 PRINT TAB(33);"Oh!X Present":PRINT
6260 LOCATE 70-LEN(DATNAM$)/2,1
6270 PRINT DATNAM$
6280 '
6290 FOR I=0 TO 4999
6300    IF INKEY$<>"" THEN 6330
6310 NEXT
6320 '
6330 CONSOLE 3,16,0,79
6340 '
6350 SCENE=1
6360 NDIR$=DRNM$
6370 '
6380 *GAME1
6390 CONSOLE 3,16,2,78
6400 DIRNB=INT(SCENE/(FILEMAX+1))+1
6410 DIRNAM$=NDIR$+STR$(DIRNB)
6420 II=0
6430 OPEN "r",#1,DRV2$+DIRNAM$+"/scene"+STR$(SCENE),80
6440 FIELD #1,76 AS DAT$
6450 '
6460 FOR I=1 TO 4
6470   GET #1,I
6480   JPT(I-1)=VAL(DAT$)
6490 NEXT
6500 FOR I=5 TO 9
6510   GET #1,I
6520   IF DAT$=SPACE$(76) THEN
6530     IF JPT(I-5)=0 THEN
6540       KYT$(I-5)=LEFT$(DAT$,16)
6550     END IF
6560   ELSE
6570     KYT$(I-5)=LEFT$(DAT$,16)
6580   END IF
6590 NEXT
6600 '
6610 CONSOLE 3,16,2,78
6620 '
6630 PAGE=1
6640 FOR I=0 TO 14
6650   GET #1,I+II+10+(PAGE-1)*15
6660   MES$(I)=DAT$
6670   IF I>0 THEN
6680     B$=LEFT$(DAT$,2)
6690     IF B$=", " OR B$=". " OR B$="。 " OR B$="、 " THEN
6700       MES$(I-1)=MES$(I-1)+B$
6710       MES$(I)=MID$(MES$(I),3)
6720       IF MES$(I)=SPACE$(LEN(MES$(I))) THEN II=II+1:GOTO 66
50
6730     END IF
6740   END IF
6750   IF EOF(#1) THEN
6760     PAGE=0
6770     GOTO 6800
6780   END IF
6790 NEXT
6800 IF I<14 THEN
6810   I=I+1
6820   MES$(I)=SPACE$(76)
6830   GOTO 6800
6840 END IF
6850 '
6860 GOSUB *WPRINT
6870 '
6880 IF PAGE<>0 THEN
6890   GOSUB *PAUSE
6900   PAGE=PAGE+1
6910   GOTO 6640
6920 END IF
6930 '
6940 IF LEFT$(KYT$(4),1)<>"*" THEN 7100
6950 GOSUB *PAUSE
6960 SCENE=JPT(0)
6970 CLOSE
6980 GOTO *GAME1
6990 '
7000 *PAUSE
7010 CONSOLE 19,2,0,79
7020 CLS
7030 PRINT TAB(24);"＊＊　Ｈｉｔ　ａｎｙ　ｋｅｙ　＊＊"
7040 CONSOLE 19,2,2,76
7050 KEY 0,""
7060 REPEAT
7070 UNTIL INKEY$<>""
7080 RETURN
7090 '
7100 CONSOLE 19,2,0,79
7110 CLS
7120 FOR I=0 TO 3
7130   PRINT SPACE$(2)+KYT$(I)+SPACE$(2);
7140 NEXT
7150 '
7160 CLOSE
7170 X=2:Y=19:CMC=2
7180 GOSUB *CUR
7190 IF DRNM$="終了" AND JPT(0)=0 THEN *START
7200 IF X=2 THEN SCENE=JPT(0)
7210 IF X=22 THEN SCENE=JPT(1)
7220 IF X=42 THEN SCENE=JPT(2)
7230 IF X=62 THEN SCENE=JPT(3)
7240 GOTO *GAME1
7250 '
7260 *WPRINT
7270 FOR I=15 TO 0 STEP -1
7280   CONSOLE 3,I+1
7290   LOCATE 0,I+3
7300   PRINT
7310
7320   CONSOLE I+3,16-I
7330   LINE (1,(I+4)*8)-(318,(I+4)*8),4
7340   PSET (1,(I+4)*8),0
7350   LINE (2,(I+4)*8+1)-(317,(I+4)*8+1),4
7360   LINE (1,(I+3)*8)-(318,(I+3)*8),15
7370   LINE (2,(I+3)*8+1)-(317,(I+3)*8+1),0
7380   PRINT MES$(I)
7390 NEXT
7400 LINE (1,(I+4)*8)-(318,(I+4)*8),4
7410 PSET (1,(I+4)*8),0
7420 LINE (2,(I+4)*8+1)-(317,(I+4)*8+1),4
7430 CONSOLE 3,16,2,78
7440 RETURN
7450 '
7460 *ERROR
7470 INIT "crt1:,,,0"
7480 CLS 3
7490 ER=0
7500 GOSUB *ERRPRT
7510 PRINT "エラー番号  = ";ERR
7520 PRINT "エラー発生行 = ";ERL
7530 CLOSE
7540 END
```

▶ドラスピ，サンダーフォースII，沙羅曼蛇，どれをとっても最高だが，R-TYPEがまだ発売されていないのは残念だ。ところでナムコの新作メタルホークはムチャクチャ面白いですよ。
横田　紀明　(21)　福岡県

ピコマゲドンへの道・完結編

満開製作所 Iwai Ippei
祝　一平

シリーズでお届けしていますこのC調言語応用編「ピコマゲドンへの道」ですが，なんと，第2回をもっていきなり完結編を迎えてしまうんだそうです。このピコマゲドンのゲームバランスについては，皆さんの精進にすべてお任せというオマケまで付いて，いつもどおりの展開が繰り広げられます。どうかごゆっくりとお楽しみください。

先月は「その壱」として，実にいーかげんなピコピコRPGを作ったわけである。で，それは6人組だったわけであるが，その後あれこれといじってるうちに，やっぱり6人もいるとごちゃごちゃしてよくないと思うようになり，今回はいきなり「Z，X，C」の3人に減らしたのであった。

たとえば，ウィザードリィなどでは6人以内でパーティを組めることになっているが，これは基本的には「職能の細分」という文化的基盤からきとるわけだ(多分)。しかし，考えてみるなら，魔法使いが長剣を振り回していけないなどとゆーのはおかしいではないか。なんでも，魔法使いは鉄を身につけていちゃいけないなんてことを，ヨーロッパ文化圏では暗黙のうちに了解しているらしいのであるが，こっちには比叡山の僧兵というものもあるのだからして，パーティの3人はおのおの剣を持ってるし，また魔法も使えるということになっている（早く言えば手抜きでもあるのだが)。

進歩したのである

さて，先月から大きく変わったのが，飛び道具を使えるようになったのと，モンスターを全滅させるとアイテムが出てくるということ，そして魔法が使えるということである。

飛び道具はとりあえずは手裏剣だけである（種類を増やしてもそれほど意味はないだろう)。最初はそれぞれ10個ずつ持っており，(当たろうが外れようが)投げるたびに1個ずつ減っていく。補充できるのはアイテムを拾ったときだけである。

アイテムであるが，これは，「＊」，「）」，「！」という3種類がある。「＊」はゴールド，「）」は手裏剣，「！」は治癒の薬である。モンスターをやっつければ一応どれかが出るようになっているが，意地悪なことに「＊」の出る確率が一番高くなっている。「）」は手裏剣で，拾ったキャラクタの持ち物となる。キャラクタ同士での物のやり取りはできないことになっているので，誰に拾わせるかはよく考えること。

また，薬も拾った瞬間に飲んでしまうので，これまた誰に拾わせるのか問題であろう。

次に魔法であるが，これは戦闘中だけにしか使えない（つまり，キャンプは張れない)。［M］キーを押すと，画面の右上に使える魔法のリスト(いまのところは火炎と治療だけ)が表示されるので，[1]か[2]で魔法を選択する。そのあと，魔法をかけるキャラクタを聞いてくるので，アルファベットを入力するのである。火炎ならモンスターのキャラクタ，治癒なら「Z,X,C」のうちのどれかである。いまのところはいーかげんなので，魔法を1回使うたびに最初10あったマジックポイントが1ずつ減っていくようになっている。

本当は，マジックポイントというのは人工的なワザとらしさがあっていやなのであるが，それよりもよい手が思いつかなかったので，とりあえずはマジックポイント制を継承している。

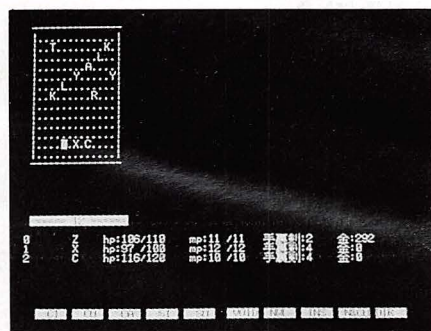
ここで反省である

移動であるが，どうやら一度に1歩ずつにしたほうがよかったように思われる。現在のように，「2歩動けて，[5]キーで移動終了」というのはうっとーしいようである。

あと，やっぱり当たり前のことであるが，

1) バラエティあふれる，歯ざわりのよいモンスターたち
2) ジューシィであと味が爽やかなアイテム
3) いつまでも飽きさせない淡泊な迷路
4) ばったりとしていて，それでいてしつこくないモンスターとの出会い
5) ふくいくとしたピット

などなどがないと，やはりもの足りないよーである。

ま，こんなところである。今月で一応目鼻は揃ったわけであるが，やはりRPGの命であるところの「ゲームバランス」がかなり

ズサンである。だからにして，好きなように改造してみていただきたい。もっとも手軽なのは経験値によるレベルアップであろう。あと，「逃げる」という選択もあったほうがいいな。それから，それから……である。

リストを見てもらうとわかると思うが，このテのプログラムでは，アイテム/魔法を1種類増やすたびに，それに対応した部分がモコモコと膨らんでくるのである。だから，全体としてはかなりとりとめのないものとなっていく。これは宿命であるから，真面目にグローバル変数の管理をしないと死ぬことになる。注意していただきたい。

なお，先月のプログラムにはバグがあったので，まともに動かない人もいたであろう。それから致命的とまではいかなかったのだが，rnd( )関数にもバグがあったようである。そのほか，あちこちでちまちまと変更してあるので注意していただきたい。

今月はぷりぷりとしたいいかげんさであった。それではまた来月。

リスト1 iyaan.h Ver. 2

```
 1: typedef struct kyara {
 2:         int X;
 3:         int Y;              /* 位置 */
 4:         int fx;
 5:         int fy;             /* フォーメーションの位置 */
 6:         char face;          /* ASDZXC */
 7:         int maxhp;          /* 体力の最大値 */
 8:         int hp;             /* その時の体力 */
 9:         int dex;            /* 機敏さ：移動能力 */
10:         int weapon;         /* 武器番号 */
11:         char foot;          /* 足の下 */
12:
13:         int maxmp;          /* 魔力の最大値 */
14:         int mp;             /* その時の魔力 */
15:         int ken;            /* 手裏剣の数 */
16:         int gold;           /* 所持金 */
17:         };
18:
19: typedef struct mons {
20:         int X;
21:         int Y;              /* 位置 */
22:         char face;          /* alphabet except ASDZXC */
23:         int hp;             /* 体力 */
24:         int dex;            /* 機敏さ：移動能力 */
25:         int weapon;         /* 攻撃の種類 */
26:         char foot;          /* 足の下 */
27:         int mp;             /* その時の魔力 */
28:         };
29:
30: #define PN      3           /* 3人編成 */
31: #define MN      20          /* 怪物は最大20匹 */
32:
33: #define XMAX    60
34: #define YMAX    20
35:
36: /* 武器 */
37: #define HAND            0
38: #define SHORT_SWORD     1
39: #define SWORD           2
40: #define LONG_SWORD      3
41: #define STAFF           4
42:
43: /* モンスターの攻撃種類 */
44: #define BITE    1           /* 咬む */
45: #define BEAT    2           /* 殴る */
46: #define CLAW    4           /* 引っ掻く */
47: #define POISON  8           /* 毒 */
48: #define BLOW    16          /* 火を吐く */
49:
50:
```

リスト2 pico.c Ver. 2

```
  1: #include        "iyaan.h"
  2:
  3: char vvm[YMAX][XMAX+1];
  4:
  5: struct kyara dp[PN] ={
  6: {       0,0, 6,13,'Z',110,110,2,SWORD   ,'.',11,11,10,0},
  7: {       0,0, 8,13,'X',100,100,2,SWORD   ,'.',12,12,10,0},
  8: {       0,0,10,13,'C',120,120,2,SWORD   ,'.',10,10,10,0}
  9: };
 10:
 11: mname[]={       "火炎",
 12:                 "治療"};
 13:
 14: #define MNUMBER 2
 15:
 16: struct mons dm[MN];
 17:
 18: int mlast;
 19:
 20: int fl;         /* 何階か */
 21:
 22: main()
 23: {
 24:         init();
 25:
 26:         for(fl=1;fl<100;fl++)
 27:         {       box(1,1,15,15);
 28:                 dispfl(fl);
 29:                 syutugen(fl);                   /* 転送 */
 30:                 mlast = genmon(fl);             /* 怪物の生成 */
 31:                 if (fight() == 0) break;        /* 生き残りが0なら終わり */
 32:         }
 33:         endinit();
 34: }
 35:
 36: int pp;         /* プレイヤー番号 */
 37: int mm;         /* モンスター番号 */
 38: char pact[PN];  /* 動いたかどうかのフラグ */
 39: char mact[MN];  /* モンスターが動いたかどうかのフラグ */
 40:
 41: left()          /* 生き残りの数 */
 42: {
 43:         register int i,p;
 44:
 45:         for(p=i=0;i<PN;i++)
 46:                 if (dp[i].hp>0)
 47:                         p++;
 48:         return(p);
 49: }
 50:
 51: aleft()         /* 動き残りの数 */
 52: {
 53:         register int i,p;
 54:
 55:         for(p=i=0;i<PN;i++)
 56:                 if ((dp[i].hp>0) && (pact[i]>0))
 57:                         p++;
 58:         return(p);
 59: }
 60:
 61: mleft()         /* モンスターの生き残りの数 */
 62: {
 63:         register int i,p;
 64: /*
 65: loc(0,29);
 66: printf("                                        ");
 67: loc(0,29);*/
 68:         for(p=i=0;i<mlast;i++) {
 69: /*      printf("%d=%3d::",i,dm[i].hp);*/
 70:                 if (dm[i].hp>0)
 71:                         p++;
 72:         }
 73:         return(p);
 74: }
 75:
 76: maleft()        /* モンスターの動き残りの数 */
 77: {
 78:         register int i,p;
 79: /*
 80: loc(0,30);
 81: printf("                                        ");
 82: loc(0,30);*/
 83:         for(p=i=0;i<mlast;i++) {
 84: /*      printf("%d=%3d::",i,dm[i].hp);*/
 85:                 if ((dm[i].hp>0) && (mact[i]>0))
 86:                         p++;
 87:         }
 88:         return(p);
 89: }
 90:
 91: fight()         /* 戦闘開始 */
 92: {
 93:         while(1) {
 94:                 fight_init();   /* set pact[],mact[] for new turn */
 95:                 while(aleft() || maleft()) {
 96:                         if (aleft()) {
 97:                                 do_man();
 98:                         }
 99:                         if (mleft() == 0) {     /* モンスターをやっつけた */
100:                                 doitem();
101:                                 return(left());
102:                         }
103:                         if (maleft()) {
104:                                 do_mon();
105:                         }
106:                         if (left() ==0)
107:                                 return(left()); /* どひー！ 全滅だぁ */
108:                 }
109:         }
110: }
111:
112: fight_init()    /* フラグのセットなどなど */
113: {
114:         int lp,i;
115:
116:         for(i=PN-1;i>=0;i--) {
117:                 if (dp[i].hp>0) {
118:                         lp = i;
119:                         pact[i] = 1;    /* active */
120:                         white(i);       /* show */
121:                 } else {
122:                         pact[i] = 0;    /* non-active */
123:                 }
124:         }
125:         pp = lp;        /* pp = fighter number */
126:
127:         for(i=mlast-1;i>=0;i--) {
128:                 if (dm[i].hp>0) {
129:                         lp = i;
130:                         mact[i] = 1;    /* active */
131:                         whitem(i);      /* show */
132:                 } else {
133:                         mact[i] = 0;    /* non-active */
134:                 }
135:         }
136:         mm = lp;        /* mm = monster number */
137: }
138:
```

```
139: do_man()         /* pp番目のキャラを操作する */
140: {
141:         int x,y,x0,y0,vx,vy;
142:         int w,d,pr;
143:         char c,fc;
144:
145: again:
146:         if ((pact[pp]==0) || (dp[pp].hp <= 0)) nextpp();
147: /* do player */
148:         x=dp[pp].X;y=dp[pp].Y;
149:         fc = dp[pp].face;
150:         loc(x,y);
151:         x0 = x;y0 = y;  /* save x,y */
152:         d = dp[pp].dex;
153:
154: mokkai:
155:         c = toupper(getch());
156:         switch(c) {
157:         case 'Z': w=0;goto foo;
158:         case 'X': w=1;goto foo;
159:         case 'C': w=2;          /* う～ん、キタナイ */
160: foo:
161:                 if (pact[w]>0) {
162:                         if ((x!=x0)||(y!=y0)&&(pact[pp]>=0)) {
163:                                 /* 前のキャラが動いていたなら */
164:                                 dp[pp].X = x;
165:                                 dp[pp].Y = y;
166:                                 pact[pp] = -1;
167:                                 blue(pp);
168:                                 pp=w;
169:                                 return; /* 一人上がり */
170:                         }
171:                         pp = w;         /* 新しい pp */
172:                         goto again;     /* やり直し */
173:                 }
174:                 break;  /* mokkai */
175:
176:         case '¥x1b':    /* cancel */
177: modosi:         if (pact[pp]>0) {
178:                         prnc(x,y,dp[pp].foot);
179:                         dp[pp].foot=getvvm(x0,y0);
180:                         prnc(x=x0,y=y0,fc);     /* 戻す */
181:                         d = dp[pp].dex;
182:                 }
183:                 break;  /* mokkai */
184:
185:         default:
186:                 if (c == 'Q') {
187:                         endinit();
188:                 }
189:
190:                 if (c == '5') {         /* 立ち止まる */
191:                         dp[pp].X = x;
192:                         dp[pp].Y = y;
193:                         pact[pp] = 0;
194:                         blue(pp);
195:                         return;         /* 一人上がり */
196:                 }       /* end 1 */
197:
198:                 if (c == 'M') {         /* MAGIC ! */
199:                         if (dp[pp].mp == 0) {
200:                                 mess("マジックポイントがありません");
201:                                 putchar(7);     /* beep */
202:                                 wait(90);
203:                                 mess0();
204:                                 break;
205:                         }
206:                         if (!cast()) goto modosi;
207:                         dp[pp].mp--;
208:                         dp[pp].X = x;
209:                         dp[pp].Y = y;
210:                         dispstat(pp);
211:                         pact[pp] = 0;
212:                         blue(pp);
213:                         return;         /* 一人上がり */
214:                 }
215:
216:                 if (c == 'T') {         /* SHOT ! */
217:                         if (dp[pp].ken == 0) {
218:                                 mess("手裏剣がありません");
219:                                 putchar(7);     /* beep */
220:                                 wait(90);
221:                                 mess0();
222:                                 break;
223:                         }
224:                         do {    mess("方向は？");
225:                                 c = toupper(getch());
226:                         } while(!((c == '¥x1b')||(isvect(c))));
227:                         mess0();
228:                         if (c == '¥x1b') goto modosi;
229:
230:                         dp[pp].X = x;
231:                         dp[pp].Y = y;
232:
233:                         c -= '0';
234:                         vx = (c+2)%3-1;
235:                         vy = -(c-1)/3+1;/* 1-9から射撃方向を計算する */
236:
237:                         dp[pp].ken--;
238:                         dispstat(pp);
239:                         pr = 20;
240:                         while((w = getvvm(x += vx,y += vy)) == '.') {
241:                                 col(32);
242:                                 prnc(x,y,')');
243:                                 col(33);
244:                                 prnc(x,y,w);
245:                                 pr--;
246:                         }
247:
248:                         if (ismonster(w) && (rnd(20)<pr)) {     /* 攻撃 */
249:                                 mess("当たった！");
250:                                 attack(pp,0,x,y);       /* 当たった */
251:                                 wait(50);
252:                                 mess0();
253:                         } else {
254:                                 mess("外れた！");
255:                                 wait(80);
256:                                 mess0();
257:                         }
258:                         pact[pp] = 0;
259:                         blue(pp);
260:                         return;
261:                 }
262:
263:                 if (isvect(c)) {
264:                         c -= '0';       /* 移動もしくは攻撃 */
265:                         vx = (c+2)%3-1;
266:                         vy = -(c-1)/3+1;/* 1-9から移動方向を計算する */
267:                         w = getvvm(x+vx,y+vy);
268:                         if ((w == '.') && (d>0)) {      /* 移動 */
269:                                 prnc(x,y,dp[pp].foot);
270:                                 x = x+vx;
271:                                 y = y+vy;
272:                                 dp[pp].foot = w;
273:                                 prnc(x,y,fc);
274:                                 d--;
275:                                 break;  /* mokkai */
276:                         }
277:
278:                         if (ismonster(w)) {             /* 攻撃 */
279:                                 dp[pp].X = x;
280:                                 dp[pp].Y = y;
281:                                 attack(pp,d,x+vx,y+vy);
282:                                 pact[pp] = 0;
283:                                 blue(pp);
284:                                 return;
285:                         }
286:                 }
287:         }
288:         goto mokkai;
289: }
290:
291: isvect(c)
292: char c;
293: {
294:         return(('1' <= c) && (c <= '9') && (c != '5'));
295: }
296:
297: do_mon()
298: {               /* 今度はモンスターの番 */
299:         int q,q0,d,d0;
300:         int px,py;
301:
302:         mm = rnd(mlast-1);      /* 0<= mm < mlast */
303:         while ((mact[mm]==0) || (dm[mm].hp <= 0)) {
304:                 mm++;
305:                 mm %= mlast;    /* 0<= mm < mlast */
306:         }       /* まだ動いてなくて、かつ生きている奴を捜す */
307:
308:         for(d0=1000,q=0;q<PN;q++) {
309:                 if (dp[q].hp<=0) continue;
310:                 d = dist(dm[mm].X,dm[mm].Y,dp[q].X,dp[q].Y);
311:                 if (d0>d) {
312:                         d0=d;q0=q;
313:                         px=dp[q].X;py=dp[q].Y;
314:                 } else if ((d0 == d) && (rnd(9)<3)) {
315:                                 d0=d;q0=q;
316:                                 px=dp[q].X;py=dp[q].Y;
317:                         }       /* 適当に散らす */
318:         }
319:         if (movem(px,py) == 1) {        /* 隣り合ったか？ */
320:                 mattack(px,py);
321:         }
322:         mact[mm] = 0;
323:         bluem(mm++);
324: }
325:
326: /* mm番のモンスターをできるだけpx,pyの近くまで動かす */
327: movem(px,py)
328: int px,py;
329: {
330:         int x,y,x0,y0,dx,dy,vx,vy;
331:         int d;
332:         int a,b;
333:         char c;
334:
335:         d=dm[mm].dex;
336:         x=dm[mm].X;
337:         y=dm[mm].Y;
338:         dx=px-x;
339:         dy=py-y;
340:
341:         while(d--) {
342:                 vx=sign(dx);
343:                 vy=sign(dy);
344:                 x0=x;
345:                 y0=y;
346:                 if (isplayer(getvvm(x+vx,y+vy))) break;
347:                                         /* プレイヤーの隣に来た */
348:                 if (getvvm(x+vx,y+vy) == '.') {
349:                         x += vx; dx -= vx;
350:                         y += vy; dy -= vy;
351:                 } else {
352:                         for(a=-1;a<=1;a++)      /* 障害物を回り込む */
353:                         for(b=-1;b<=1;b++)      /* ううう、入れ子が… */
354:                         if ((getvvm(x+a,y+b)=='.')&&
355:                         ((dist(x+a,y+b,px,py)<=dist(x,y,px,py)))) {
356:                                 x += a;
357:                                 y += b;
358:                                 a = b = 3;
359:                         }
360:                 }
361:                 if ((x0!=x)||(y0!=y)) { /* 動いたならば書き変える */
362:                         prnc(x0,y0,dm[mm].foot);
363:                         dm[mm].foot = getvvm(x,y);
364:                         prnc(x,y,dm[mm].face);
365:                 }
366:         }
367:         dm[mm].X = x;
368:         dm[mm].Y = y;
369:         return(dist(x,y,px,py));
370: }
371:
372: sign(x) /* 符合により、－１，０，１を返す */
373: int x;
374: {
375:         if (x>0) return(1); else if (x<0) return(-1); else return(0);
376: }
377:
378: nextpp()        /* 次のキャラクタ番号 */
379: {
380:         if (aleft()>0)
381:                 do {    if (++pp >= PN) pp=0;
382:                 } while(pact[pp]<=0);
383: }
384:
385: int dist(x1,y1,x2,y2)   /* ２点間の距離 */
386: int x1,y1,x2,y2;
387: {
388:         int w;
389:         if ((x1 -= x2)<0) x1 = -x1;
390:         if ((y1 -= y2)<0) y1 = -y1;
391:
392:         return((x1>y1)? x1:y1);
```


▶12月号の169ページを読んで，私は感動してしまった。編集室にも阪急ファンがいたのですね。もしよかったら，誰のファンなのか教えてください。ちなみに私は松永のファンだが，最近は藤井を押しています。　　芳野　達郎　(18)　東京都

```
393: }
394:
395: mattack(px,py)  /* モンスターによる攻撃 */
396: int px,py;
397: {
398:         int i,j,power;
399:         char c;
400:
401:         c = getvvm(px,py);
402:         for(i=0;i<PN;i++)
403:                 if (dp[i].face == c) break;
404:
405: /* i番キャラクタを攻撃する */
406:         power = dm[mm].hp/3;
407:         power = rnd(power);     /* 攻撃力の算定 */
408:         col(36);                /* リバース */
409:         for(j=0;j<10;j++) {     /* ピクピク */
410:                 prnc(dp[i].X,dp[i].Y,tolower(dp[i].face));
411:                 prnc(dp[i].X,dp[i].Y,dp[i].face);
412:         }
413:         col(33);
414:
415:         if ((dp[i].hp -= power)<=0) {
416:                 dp[i].hp=0;
417:                 pact[i]=0;
418:                 prnc(dp[i].X,dp[i].Y,dp[i].foot);
419:         } else if (pact[i]) white(i); else blue(i);
420:         dispstat(i);
421: }
422:
423: attack(p,d,x,y)
424: int p,d,x,y;
425: {
426:         int m,j;
427:         int power;
428:
429:         for(m=0;m<mlast;m++)
430:                 if ((x==dm[m].X)&&(y==dm[m].Y)&&(dm[m].hp>0)) break;
431:
432:         if (m == mlast) {       /* デバッグ用のトラップ */
433:                 for(m=0;m<mlast;m++) {
434:                         printf("%d:x=%d,y=%d¥n",m,dm[m].X,dm[m].Y);
435:                 }
436:                 getch();
437:         }
438:
439:         power = dp[p].hp/3+d;
440:         power = rnd(power)+1;
441:         dm[m].hp -= power;
442:         col(36);
443:         for(j=0;j<10;j++) {     /* ピクピク */
444:                 prnc(dm[m].X,dm[m].Y,tolower(dm[m].face));
445:                 prnc(dm[m].X,dm[m].Y,dm[m].face);
446:         }
447:         col(33);
448:
449:         if (dm[m].hp <= 0) {
450:                 dm[m].hp=0;
451:                 mact[m]=0;
452:                 if (mleft()) prnc(dm[m].X,dm[m].Y,dm[m].foot);
453:                 else putitem(dm[m].X,dm[m].Y);
454: /*              putchar(7);*/
455:         } else if (mact[m]) whitem(m); else bluem(m);
456: }
457:
458: init()
459: {
460:         screen(1,0,1,0);
461:         cls();
462: }
463:
464: endinit()
465: {
466:         screen(2,0,1,0);
467:         exit();
468: }
469:
470: blue(p)         /* 動き終わったら青くする */
471: int p;
472: {       int x,y;
473:
474:         col(31);
475:         prnc(dp[p].X,dp[p].Y,dp[p].face);
476:         col(33);
477: }
478:
479: bluem(m)        /* 動き終わったら青くする */
480: int m;
481: {       int x,y;
482:
483:         col(31);
484:         prnc(dm[m].X,dm[m].Y,dm[m].face);
485:         col(33);
486: }
487:
488: white(p)
489: int p;
490: {       int x,y;
491:
492:         col(33);
493:         prnc(dp[p].X,dp[p].Y,dp[p].face);
494: }
495:
496: whitem(m)
497: int m;
498: {       int x,y;
499:
500:         col(33);
501:         prnc(dm[m].X,dm[m].Y,dm[m].face);
502: }
503:
504: isplayer(c)
505: char c;
506: {
507:         switch(toupper(c)) {
508:         case 'Z':
509:         case 'X':
510:         case 'C':
511:                 return(1);
512:         default:
513:                 return(0);
514:         }
515: }
516:
517: ismonster(c)
518: char c;
519: {
520:         if (isplayer(c)) return(0);
521:         return(isalpha(c));
522: }
523:
524: isitem(c)
525: char c;
526: {
527:         switch(c) {
528:         case '!':       return(1);
529:         case ')':       return(2);
530:         case '*':       return(3);
531:         default:        return(0);
532:         }
533: }
534:
535: int genmon(f)   /* モンスターの生成 */
536: int f;
537: {
538:         int i,m;
539:
540:         m = 0;
541:         genmon1(f,m++); /* 最低1匹は出る */
542:         for(i=0;(i<f)&&(m<20);i++)
543:                 if (rnd(f)&1)
544:                         genmon1(f,m++);
545:         return(m);
546: }
547:
548: /*              "BEHIKSORZCLAQNYTWFPUGMVXDJ";   */
549: char mlist[] =  "BEHIKSORKQLAQNYTWFPUGMVGDJ";
550:
551: char mpow[] = { 2,2,8,2,4,3,8,6,
552:                 8,12,2,0,8,0,12,28,
553:                 6,9,16,33,39,34,10,12,
554:                 16,32};
555:
556: genmon1(f,n)
557: int f,n;
558: {
559:         int i,p,min,max;
560:         int x,y;
561:         char c;
562:
563:         if (f > 25) f = 25;
564:
565:         if (f <= 10) {
566:                 min = 0;
567:                 max = f + 3;
568:         } else if (f <= 19) {
569:                 min = f - 6;
570:                 max = f + 3;
571:         } else {
572:                 min = f - 6;
573:                 max = 25;
574:         }
575:
576:         do {
577:                 x = 2+rnd(13);
578:                 y = 2+rnd(6);
579:         } while(getvvm(x,y)!='.');
580:         dm[n].X = x;
581:         dm[n].Y = y;
582:
583:         i = min + rnd(max-min);
584:         c = mlist[i];
585:         p = mpow[i];
586:         prnc(x,y,c);
587:
588:         dm[n].face = c;
589:         dm[n].hp = 10+p*3+rnd(p);
590:         dm[n].dex = 2+(p/10);
591:         dm[n].weapon = BEAT;
592:         dm[n].foot = '.';
593:         dm[n].mp = 0;
594: }
595:
596: syutugen(fl)    /* 新しい部屋へ */
597: int fl;
598: {
599:         int p;
600:         int i,j,w;
601:
602:         for(p=0;p<PN;p++)
603:         {       i = dp[p].X = dp[p].fx;
604:                 j = dp[p].Y = dp[p].fy;
605:                 w = dp[p].hp;
606:                 if (w<dp[p].maxhp)
607:                         dp[p].hp += (dp[p].maxhp - w)/2;/* hpの回復 */
608:                 w = dp[p].mp;
609:                 if (w<dp[p].maxmp)
610:                         dp[p].mp += (dp[p].maxmp - w)/2;/* mpの回復 */
611:                 dp[p].foot = getvvm(i,j);
612:                 prnc(i,j,dp[p].face);
613:                 dispstat(p);
614:         }
615: }
616:
617: dispstat(p)     /* キャラクタの状態を表示する */
618: int p;
619: {
620:         loc(0,23+p);
621:         printf("%d¥t%c    hp:%-3d/%-3d    mp:¥
622: %-3d/%-3d  手裏剣:%-3d  金:%-3d",
623:         p,dp[p].face,dp[p].hp,dp[p].maxhp,dp[p].mp,
624: dp[p].maxmp,dp[p].ken,dp[p].gold);
625: }
626:
627: getvvm(x,y)
628: int x,y;
629: {
630:         return(vvm[y][x]);
631: }
632:
633: box(x1,y1,x2,y2)
634: int x1,y1,x2,y2;
635: {
636:         int i,j;
637:         char bar[81];
638:         char roo[81];
639:
640:         if ((j = x2-x1+1)>80) return;
641:         for(i=0;i<j;i++)
642:         {       bar[i]='-';
643:                 roo[i]='.';
644:         }
645:         bar[i]=roo[i]='¥0';
646:         roo[0]=roo[i-1]=';';
```

▶とうとうX68000用MIDIボードが発売された。これでミュージシャンの方々にX68000を使ってもらって，テレビに出てもらう。するとアマチュアの人も使うようになって，ユーザーも増えるのだ。　本井　進　(19)　石川県

```
647:
648:         prn(x1,y1,bar);
649:         for(i=0;i<y2-y1+1-2;i++)
650:         {       prn(x1,y1+i+1,roo);
651:         }
652:         prn(x1,y2,bar);
653: }
654:
655: dispfl(f)       /* 何階か表示する */
656: int f;
657: {       col(7);
658:         loc(1,21);
659:         printf("====== %2d ======",f);
660:         col(33);
661: }
662:
663: prn(x,y,s)
664: int x,y;
665: char *s;
666: {
667:         register char *p;
668:
669:         loc(x,y);
670:         printf("%s",s);
671:
672:         p = &vvm[y][x];
673:         while(*s)
674:                 *p++ = *s++;
675: }
676:
677: prnc(x,y,c)
678: int x,y;
679: char c;
680: {
681:         loc(x,y);
682:         printf("%c",c);
683:         vvm[y][x] = c;
684:         loc(x,y);
685: }
686:
687: /* display vvm */
688: flush()
689: {
690:         int i;
691:
692:         for(i=0;i<YMAX;i++)
693:         {       loc(0,i);
694:                 printf("%s",vvm[i]);
695:         }
696: }
697:
698: loc(x,y)
699: {
700:         printf("¥x1b=%c%c",y+0x20,x+0x20);      /* locate x,y */
701: }
702:
703: cls()
704: {
705:         register int x,y;
706:
707:         printf("¥x1b[2J");                      /* cls "con:" */
708:         for(y=0;y<YMAX;y++)
709:         {       for(x=0;x<XMAX;x++)
710:                         vvm[y][x]=' ';  /* clear vvm */
711:                 vvm[y][x]='¥0';         /* set EOS */
712:         }
713: }
714:
715: unsigned int seed = 0xe933;     /* 初期値はe933H */
716:
717: int rnd(max)
718: int max;
719: {
720:         seed = 899*seed;
721:         return(seed % (max+1)); /* return 0-max */
722: }
723:
724: mess(s)
725: char *s;
726: {       prn(0,17,s);
727: }
728:
729: mess0()
730: {       mess("                                        ");
731: }
732:
733: col(c)
734: int c;
735: {
736:         char *s[5];
737:
738:         itoa(c,s,10);
739:         printf("¥x1b[%sm",s);
740: }
741:
742: wait(i)
743: int i;
744: {       int a;
745:
746:         a = ONTIME();
747:         while(((ONTIME()-a+8639999) % 8639999)<i);
748: }
749:
750: cast()  /* 魔法処理 */
751: {
752:         int c,f,m,m0,p,w;
753:
754:         mmenu();
755:         mess("魔法の番号は？");
756:         c = toupper(getch());
757:         mess0();
758:         mmenu0();
759:         c -= '1';
760:         if (c>MNUMBER) return(0);       /* mokkai */
761:         mess("誰に対してですか？");
762:         f = toupper(getch());
763:         mess0();
764:         if (! isalpha(f)) return(0);    /* mokkai */
765:
766:         switch(c) {
767:         case 0: /* 火炎 */
768:                 for(m=0;m<mlast;m++) {
769:                         m0 = rnd(mlast-1);      /* ランダムに抽選 */
770:                         if ((dm[m0].face == f)&&(dm[m0].hp>0)) {
771:                                 mess("攻撃成功！");
772:                                 attack(pp,5,dm[m0].X,dm[m0].Y);
773:                                 wait(50);
774:                                 mess0();
775:                                 return(1);
776:                         }
777:                 }
778:                 mess("攻撃失敗・・・");
779:                 wait(80);
780:                 mess0();
781:                 break;
782:
783:         case 1: /* 治癒 */
784:                 for(p=0;p<PN;p++) {
785:                         if ((dp[p].face == f)&&(dp[p].hp>0)) {
786:                                 w = dp[p].maxhp - dp[p].hp;
787:                                 w = 1 + rnd(w/3);
788:                                 dp[p].hp += w;
789:                                 mess("ああ、甘露甘露・・・");
790:                                 wait(80);
791:                                 mess0();
792:                         }
793:                 }
794:                 break;
795:         }
796:         return(1);
797: }
798:
799: mmenu()
800: {       int i;
801:         for(i=0;i<MNUMBER;i++) {
802:                 prnc(18,i+2,'1'+i);
803:                 prn(20,i+2,mname[i]);
804:         }
805: }
806:
807: mmenu0()
808: {       int i;
809:         char *s;
810:         s="              ";
811:         for(i=0;i<MNUMBER;i++) {
812:                 prn(18,i+2,s);
813:         }
814: }
815:
816: doitem()        /* pp番目のキャラを操作する */
817: {
818:         int x,y,x0,y0,vx,vy;
819:         int w,d,pr;
820:         char c,fc;
821:
822:         fight_init();
823:         pp = 0;
824: again1:
825:         if (dp[pp].hp <= 0) nextpp();
826:         x=dp[pp].X;y=dp[pp].Y;
827:         fc = dp[pp].face;
828:         loc(x,y);
829:         x0 = x;y0 = y;  /* save x,y */
830:
831: mokkai1:
832:         c = toupper(getch());
833:         switch(c) {
834:         case 'Z': w=0;goto fool;
835:         case 'X': w=1;goto fool;
836:         case 'C': w=2;          /* う～ん、キタナイ */
837: fool:
838:                 if (dp[w].hp>0) {
839:                         dp[pp].X = x;
840:                         dp[pp].Y = y;
841:                         pp=w;
842:                         goto again1;    /* やり直し */
843:                 }
844:                 break;  /* mokkai */
845:
846:         default:
847:                 if (c == 'Q') {
848:                         endinit();
849:                 }
850:
851:                 if (isvect(c)) {
852:                         c -= '0';       /* 移動もしくは攻撃 */
853:                         vx = (c+2)%3-1;
854:                         vy = -(c-1)/3+1;/* 1-9から移動方向を計算する */
855:                         w = getvvm(x+vx,y+vy);
856:
857:                         if (w == '.') { /* 移動 */
858:                                 prnc(x,y,dp[pp].foot);
859:                                 x = x+vx;
860:                                 y = y+vy;
861:                                 dp[pp].foot = w;
862:                                 prnc(x,y,fc);
863:                                 break;  /* mokkai */
864:                         }
865:
866:                         switch(w) {
867:                         case '*':
868:                                 dp[pp].gold += 10+rnd(fl*10);
869:                                 goto getitem;
870:                         case ')':
871:                                 dp[pp].ken += 5;
872:                                 goto getitem;
873:                         case '!':
874:                                 dp[pp].hp += 5+rnd(fl);
875: getitem:                        prnc(x,y,dp[pp].foot);
876:                                 prnc(x+vx,y+vy,fc);
877:                                 dispstat(pp);
878:                                 return;
879:                         default:        break;
880:                         }
881:                 }
882:         }
883:         goto mokkai1;
884: }
885:
886: putitem(x,y)
887: int x,y;
888: {       int r;
889:         char c;
890:
891:         r = rnd(99);
892:         if (r<60) c ='*';
893:         else if (r<90) c ='!';
894:         else c =')';
895:
896:         col(32);
897:         prnc(x,y,c);
898:         col(33);
899: }
900:
```

Z80マシン語
ゲーム工房

第6回

ついに敵機来襲

Murata Toshiyuki 村田 敏幸

基礎からスタートしたこのゲーム工房も，とうとうデラックスなカラー画面（目次参照）に，バシバシ弾を撃ちながら敵機の襲撃へと突入します。今月は特に，リスト内に細かい解説が加えられていますから，そちらも参照しながら残された課題をじっくりと学んでみてください。

さて，大詰めだ。今月は，敵も弾が撃てるようにし，背景のスクロールと合体させ，ドサクサまぎれに色もつけちゃって，一応ゲームらしい格好にするところまでをやる。とは言うものの，自機周りの当たり判定が付いていないので，自機は敵に当たろうが撃たれようが障害物に体当たりしようが決して死なないという，いわゆる無敵状態になっている。

前回まではパーツごとに作成・合体を繰り返してきたわけだが，いい加減，収拾がつかなくなってきたから，今回は最初にまとまったリストを示し，その部分部分を解説するという親切設計で話を進めることにした。リスト1～4が順にMZ-700/2000/2500，X1用の各機種に依存する部分で，リスト5が共通部だ。各機種用部の後ろに共通部をくっつけてアセンブルしてもらいたい。ただし，X1turboではX1用の一部をリスト中に示したように変更して使う。また，MZ-700の場合は「共通部」に若干の変更点があるので注意すること。

おっとそれから，リストが大きくなったため，オブジェクトの生成アドレスを変更してある。オブジェクトは4E00Hから生成されるから（注1），

＃J4E00

で実行だ。

注1）標準状態のZEDAでは4E00Hからソースプログラムを置くようになっている。そのアドレスからオブジェクトを生成するということはソースを破壊しながらアセンブルするということだ。これはオンメモリのアセンブラで大きなソースを作るときの常套手段だ。なお，この場合，アセンブル後にソースを修正するには，ロードするところから始めなければならないことは言うまでもない。

ポーズ処理：宿題の解答

まず，前回の宿題だった一時停止処理から片づける。このように一見簡単そうな処理にも，実際にプログラムにしてみてやっと気づく落とし穴があるものだ。自分で試してみた人にはもうわかっていると思うが，順に考えてみよう。

最初に一時停止の条件であるトリガー2が押されているかどうかをチェックする。このチェックの仕方は前回やった。そして，トリガー2が押されていれば，一時停止処理本体に分岐し，一時停止解除の条件が満たされるまで待つわけだ。一時停止はトリガー1か2が押されたら解除と決めてあったから，ポーズ処理ルーチンは，

1）キー入力データを得る
2）トリガー1か2が押されているかどうかを調べ，押されていなければ1）に戻る
3）どちらかが押されていたらメイン処理に復帰する

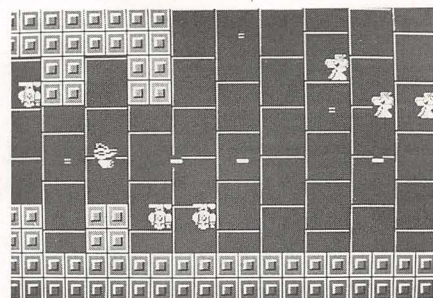
という形が考えられる。

ところが，このとおりプログラムを書いて走らせてみると，どういうわけか一時停止しない場合があるはずだ。何度か試してみると，トリガー2をチョンと押したときには一時停止することもわかると思う。これは，人間とプログラムの処理速度の違いからくるものだ。つまり，トリガー2が押され，1）の処理に入った時点で「まだトリガー2が押されている」と2)，3)のチェックに引っかかって，すぐに処理を抜けてしまうわけだ。これを避けるためには，1)の直前に「トリガー2が離されるのを待つ」処理を付け加えればよい。

リスト5の846～859行がポーズ処理サブルーチンだから，一応目を通しておいてもらおう。

敵の反撃

次に敵に弾を撃たせてみる。しかし，ただ水平に撃つだけでは面白くないので，前にさらっと書いたように敵の弾は自機がその瞬間にいる位置目がけて飛ぶようにする。考え方は2種類ほどある。

ひとつは敵の移動のときにやった，向きコードを利用する方法だ。弾は8方向（もちろん16方向などに拡張してもよい）にしか移動できないものとし，発射のときに一

番近い方向を選び，あとはその方向に直進する。この方法では自機を正確に狙うというわけにはいかないが，向きコードを増減することで，進路を変化させることができるので，「最初はちょっとずれた方向を狙って，徐々に曲げる」とか「弾がぐねぐねと曲線を描きながら飛んでいく」といった細工はしやすい。

もうひとつはグラフィックの直線描画のアルゴリズムを利用する方法だ。これにより弾は正確に自機の位置を狙って飛ぶようになる。テキストベースだと軌跡はガタガタしたものになるが，まぁ，仕方のないところだろう。目新しさを狙って，制作中のゲームではこちらを採用する。

さて，強引な話の展開だが，ここでライン描画のアルゴリズムを説明する。ライン描画の方法にもいろいろあるわけだけど，ここでは比較的オーソドックスなアルゴリズムを用いることにした。このアルゴリズムは「画面に描きうる最良の直線を描くわけではない」が，比較的処理が軽いので，パソコンのBASICなんかは大抵このアルゴリズムを使っていると思われる。

話を簡単にするために，

X1＜＝X2
Y1＜＝Y2
X2－X1＞Y2－Y1

を満たすような2点（X1，Y1）と（X2，Y2）を結ぶ場合を例に取り上げる。つまり，X軸と交わる角度が45度よりなだらかな傾きの直線だ。なお，

DX＝X2－X1
DY＝Y2－Y1

とおく。

始点から終点に達するまでの間にX座標はDX，Y座標はDYだけ増加する。いまはDX>DYなので，毎回X座標を1増やしつつ，たまにY座標を1増やせば，望む直線が描けるだろう。この「たまに」というのがくせものだが，X座標をDX回増加する間に，Y座標をDY回増やすわけだから，Y座標を増やす確率（というか頻度）はDY/DX(<=1)だということがわかる。たとえばDX=4でDY=2であればDY/DX=1/2，つまり，Xを2回増やすごとにYを1回増加させればよい。

この考え方をもう少しプログラムっぽく言い換えてみよう。次のようになる。

「初期値0の変数Pを用意し，これに点をひとつ打つごとにDY/DXを足して，Pが1以上になったらYをひとつ増やす」

この言い換えが正しいかどうか確かめるには，実際に簡単な例で試してみることだ。DX=4，DY=2の場合について考えてみると，

1) DY/DX=1/2。また，最初だから，P=0
2) 始点に点を打つ
3) Xをひとつ増やし，Pに1/2を足す。結果は1/2だから1より小さい。だからYはそのまま変化させない
4) 3)で求めた位置に点を打つ
5) またXをひとつ増やし，Pに1/2を足す。その結果Pは1以上になったのでYを1増やす。この時点で次に備えてPから1を引いておく
6) 点を打つ
7) Xをひとつ増やし，Pに1/2を足す。結果は1/2だからYはそのまま
8) 点を打つ
9) Xをひとつ増やし，Pに1/2を足す。その結果Pは1以上になったのでYを1増やす
10) 点を打つ

というように，ちゃんとXを4つ増やす間にYを2つ増やすことができた。DY/DXがもっと中途半端な値のときもうまくいくことを確認してもらいたい。以上がこのアルゴリズムの大筋だ。

では次に，

X1<=X2
Y1<=Y2
X2−X1<Y2−Y1

のような，傾きが急な直線を描くことを考えてみよう。この場合DY>DXなので，毎回Yを1増やしながらDX/DYの確率でXに1を足せばよい。いままでの例のXとYをすべて逆にしたのと同じことだ。なら，

X1>=X2
Y1<=Y2
X1−X2>Y2−Y1

の場合はどうだろう。この場合は毎回Xを1減らしつつ，DY/DXの確率でYを増やしていけばよい。さっきの例とはX座標を毎回1増やすか1減らすかの違いだ。結局，

DX=ABS（X2−X1）
DY=ABS（Y2−Y1）
SX=SGN（X2−X1）
SY=SGN（Y2−Y1）

とおいて，

1) DX>=DYならば，
Xに毎回SXを足しつつ，DY/DXの確率でYにSYを足す
2) DY>=DXならば，
Yに毎回SYを足しつつ，DY/DXの確率でXにSXを足す

というようにまとめられる。

以上のアルゴリズムを利用して，弾の発射・移動ルーチンを作成する。ここで，問題になるのはDY/DX（またはDX/DY）を求めるために割り算ルーチンが必要なことと，その結果が小数の値をとるということだ。もちろん，マシン語で実数演算ルーチンを作ることもできるわけだが，それでは処理が遅くなってしまうので，整数化して計算することを考える。たとえばDY/DXを256倍しておき（注2），「1を越えたかどうか」調べる代わりに256を越えたかどうかで判断すればよい。ここで256倍する処理はわざわざ掛け算ルーチンを作るまでもないことは，以前の話を思い出してもらえればわかると思う。残る整数の割り算に関しては囲み記事を参照してほしい。

リスト5の637行からが敵弾の発射処理ルーチンだ。最初に敵弾のデータバッファの空きを探し（バッファの内部構成は33行以下を参照），弾の初期位置となる「弾を撃つキャラクタの座標と表示アドレス」をそのままコピーしているのがわかるだろう。この先がライン描画のアルゴリズムを利用している部分だ。

ここではDX，DY，SX，SY，さらにDY×256/DX（またはDX×256/DY）を計算し，敵弾バッファ内にセットするという処理を行っている。「確率」が1を越えたときと越えなかったときの座標変位を別々に用意するなど，先に計算できるものはみんな計算しておくという方針で書いてある。ここまで計算したうえで，とどめに678行以下で，座標の変位から表示アドレスの変位を2通りの場合について求め，バッファ内に格納している。

これだけ先に計算しておくと，敵弾の移動処理はずいぶん簡単になる。サブルーチンMOVEEMSLの756行以下がその処理なんだが，「確率」にステップ値を加え，結果が256を越えたかどうかで，2種類用意した座標・表示アドレス変位を使い分けるだけですんでしまっている。ここで，「256を越えたかどうかを調べ，256以上になったら256を引く」という処理が「8ビット加算1発と，結果のキャリフラグ」だけで実現されていることにも注目してもらいたい。

あと，敵の弾はあまり速いと困るので，750行からの3行で，カウンタをチェックし，メインループ2回に1回しか進まないように調節している。EMSLCNTというワークエリアはメインルーチンの最初のほうでAAHという値で初期化してあり，これをRRC

マシン語による整数除算

10進数の割り算，たとえば123/5を筆算で行うときのことを考えてみてもらいたい。最初に被除数の最上位桁を持ってくる。いまの場合1だ。これは5では割れないから100の桁の商には0が立つ。次に被除数をもう1桁持ってきて，さっきの1と連結する。すると12になる。今度は5で割ると2が立ち，余りは2だ。この余りとさらに被除数から持ってきた3を連結すれば23だ。4が立ち，余りは3となる。最終的に24…3という答えが得られる。

さて，いきなり算数を始めたのにはわけがある。というのも，マシン語での割り算は2進数での筆算を考えて，そのままプログラムにすればよい。しかも，10進数では演算の途中で「5が立つかな？ 駄目なら4でどうだ」という勘と試行錯誤が必要だったわけだが，2進数では1が立つか0かの2通りしかないわけで，これは「引けるか引けないか」に置き換えて考えることができるという利点がある。

例としてはあまりよくないのだが，リスト5のサブルーチンDIVはHL÷Cを求めて，商をHLに余りをAレジスタに返すサブルーチンだ。速度を追求し最適化してあるためにCは127以下（最上位ビットが0）でなければならない。HLを1ビット左にシフトし，はみ出したビットをAに取り込んで，試しにCを引いてみて，仮に1を立て，キャリが立ったら（引けなかったら）引きすぎたCを足すことでAの値を元に戻し，立てた1を下ろしている。こういった処理を16回（被除数の桁数分）繰り返すと商が求まる。

なお，このサブルーチンでは被除数を格納しているHLの下のほうのビットから徐々に商が侵入してきて，16ビットシフトした時点でいつの間にかHLに商が入っているというヒネた作りになっている。HLとCにいろいろな数を入れて，Z80になりきったつもりで処理を追ってみると面白いと思う。

でローテートすると，AAH→55H→AAH……と変化し続けるだろう。AAH→55Hと変化するときはノンキャリだが，55H→AAHと変化するときにはキャリが立つので，これをチェックすれば，立派にカウンタとして機能するわけだ。この手法はリストのあちこち使われている。

注2） DYをDXで整数除算してから256倍するのでは割り算をした時点で商が0になってしまうので，DYを256倍してからDXで割るようにする。

そして背景スクロール

以前，理論だけをやった背景のスクロールをいよいよゲームに導入する。

スクロールは全画面分の絵を帯状につなげたデータを用意し，その一部を切り出して表示することで行えるんだった。いま作っているゲームは横40×縦24だから，40×24バイトのデータがあれば1画面が埋まる。これを100個つなげたものを用意すれば100画面分のスクロールが行える。が，40×24＝960バイト≒1Kバイト，100画面分用意すればなんと100Kバイトだ。これはZ80が直接扱えるメモリ量を越えている。もちろん，サンプルとして100画面分ものデータを用意するつもりはないが，そのくらいまでは拡張できるように，器は大きくとっておきたい。となれば取るべき道はひとつ。データ量を減らす工夫をすることだ。

そこで，何文字かのブロックで背景を表すことを考える。2×2とか3×3の大きさで背景の「パーツ」を用意し，これに順に番号を振っておく。そして，この番号でマップデータを用意するわけだ。今月のリストでは背景の1ブロックを4×4文字の大きさにしてある。これならデータ量は16分の1ですむから，100画面分ぐらいわけなく入るだろう。

さて，背景の単位をこのように大きくとったことで不安を覚える読者の方もいるかもしれない。パーツが大きくなると，スクロールが粗くなってしまいそうに見えるだろう。しかし，ご心配なく。ちゃぁんと手は打ってある。ポイントは「実画面よりひと回り大きい仮想画面」だ。制作中のゲームでは，40×24の実画面に対して48×32の大きさの仮想画面を用意してあり，仮想画面の(4,4)が実画面の(0,0)に対応している。実画面の周りには表示はされないが，キャラクタ4つ分のゆとりがあるわけだ。ここをうまく使う。

最初はまず仮想画面の(4,4)を左上隅とみなして，横11×縦6個の背景パーツをびっしり並べる。実画面には横10×縦6個しか表示できないから，横に11個並べるのは無駄なようだが，とりあえず気にしないでもらおう。例によって，この時点ではディスプレイに表示されるはずもなく，あとで仮想画面からごっそり実画面に転送した時点で実際の表示が行われる。もちろんこのときは無駄に書き込んだ右端の1列はディスプレイには現れない。

次に仮想画面の(3,4)を左上隅とみなして，横11×縦6個の背景パーツを並べる。そして，いつものとおりに(4,4)から(27,43)の範囲だけを実画面に転送する。もう話が見えてきたと思うが，これで背景は横に1キャラクタ分スクロールしたわけだ。以後，起点の位置を(2,4)，(1,4)とずらしながら同様の処理を繰り返し，そして，4回目のスクロールのときに，起点位置を(4, 4)に戻して，マップデータを横にひとつ進める。以下，延々と続ければ，1キャラクタ単位の横スクロールが実現される。

ジャンプテーブル

ジャンプテーブルは，BASICで言うON～GOTOに当たるような処理をするのに用いられる。その実態は，

```
DEFW    分岐先0のアドレス
DEFW    分岐先1のアドレス
DEFW    分岐先2のアドレス
DEFW    分岐先3のアドレス
          ⋮
```

のような形をした配列状のデータ列だ。たとえばAレジスタが0なら分岐先0へ，1なら分岐先1へ……というように，処理を振り分けるときに利用すると，プログラムをコンパクトにまとめることができる。

これはどうやるのかというと，分岐のキーとなる値（上ではAレジスタ）を添え字とみなし，ちょうど2バイト型の配列アクセスのようにしてレジスタペアに分岐先アドレスを取り出し，そこへジャンプすればよい。

たとえば，BCレジスタにアドレスを取り出したとすると，

```
PUSH    BC
RET
```

で，「BCの指すアドレス」へジャンプできる。また特に，「HLの指すアドレス」へジャンプする命令として，

```
JP      (HL)
```

がある。

リスト5ではイベントの処理と，敵の種類による処理ルーチンの振り分けにジャンプテーブルが用いられている。このように作っておくことで，将来の拡張が容易にできるようにしているわけだ。

リスト5の278行以下が背景を仮想画面に展開するサブルーチンだ。MAPPNTにマップデータへのポインタが，HOMEに仮想画面上の起点位置アドレスがそれぞれ格納してあり，下位サブルーチンEXMAP0で左から右，上から下へ背景パーツを並べていく。

マップデータ自体は，921行以下にとりあえず3画面分（そのうち最初と最後のブロックはわけあって同じもの），また，背景パーツのパターンは機種別部分のBPAT0以下に6つほど用意してある。データの作り方の詳しい話は来月になるけれど，リストを参考にすれば，手を加えるのも難しくはないだろう。

色をつける：MZ-700編

今月の目玉はお待ちかねのカラー化だ。まず，MZ-700からいってみよう。MZ-700にはPCGこそないが，文字単位で文字色と背景色を自由に設定できるので，ちょっと凝ってみせればゲーム画面はそれなりに華やかなものになる。

文字/背景色の指定はアトリビュートVRAMにデータを書き込むことで行う。以前話したようにメモリのD800H～DFFFHがアトリビュートVRAMになっており，表示位置に対応したアトリビュートVRAMに文字色×16＋背景色を書き込めば色の設定ができる。というわけで，新しく覚えてもらうことはなにもないといってよい。あとはプログラミング上の問題だ。

前回までは，あらかじめアトリビュートVRAMを文字色緑，背景色黒で初期化しておき，ゲーム中はテキストVRAMへの書き込みだけですませてきたけれど，文字単位で色をつけるとなると，毎回アトリビュートVRAMも更新しなければならない。これに対応するために，MZ-700では特に仮想アトリビュートVRAMなるものを導入する。これは言ってみれば仮想VRAMのアトリビュート版だ。

自機なり敵キャラなりを表示するときは，仮想VRAMにディスプレイコード，同位置の仮想アトリビュートVRAMに色コードを書き込むようにし，すべてのキャラクタの（仮想画面への）表示がすんだら，仮想VRAMと仮想アトリビュートVRAMの内容を実VRAM，実アトリビュートVRAMにそれぞれ転送することで，ディスプレイへの表示を行うわけだ。要するに，いままでテキストVRAMに対して行っていたことを，アトリビュートVRAMに対しても行うよう

にする。

ただ，これによりゲーム全体の速度はどうしても落ちてしまう。また，テキストVRAMとアトリビュートVRAMにデータを書き込む時間差が原因で，色ずれが起こることがあるという問題もある。いろいろと制限を加えて高速化することは可能なのだが，今回は多少のチラツキには目をつぶってもらいたい。

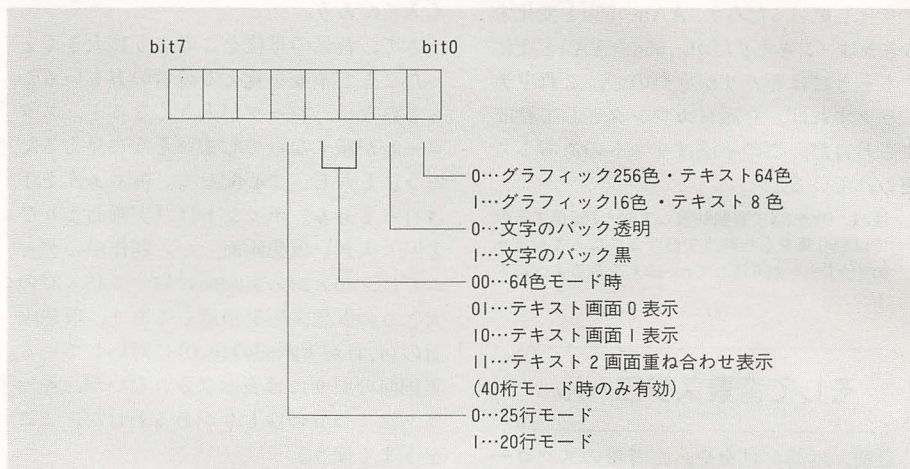
あ，忘れるところだったけど，リスト1のサブルーチンINIT700の先頭には謎のOUT命令が置かれている。これは，D000H以降をテキストVRAMにバンク切り換えするための処置で，本来この処理なしにはテキストVRAMにアクセスできない。通常，すでにテキストVRAMをアクセスできる状態になっている（はず）なので，わざわざダメを押すまでもないのだけども，縁起ものということで追加しておいた。

色をつける：MZ-2000/2200/2500編

まずMZ-2000/2200。残念なことにMZ-2000でのテキスト画面は全画面単位でしか色を設定することができない。すでに話してあるようにI/OポートのF5Hにカラーコードを OUTすることで，すべての文字の色が一斉に変わる。また，ポートF4Hで画面の背景色を設定することもできる。この2つのポートを操作して，リスト2では青地に白という設定にしてみた。

ついでに画面モードの設定の仕方も話しておく。MZ-2000では画面の横幅の40/80桁の切り換えを，ポートE8Hの第5ビットで切り換える。このビットが0なら40文字，1なら80文字となる。ポートE8Hはバンク切り換えとキー入力にも使われているので，ほかの機能の設定を変えてしまわないよう，一度ポートを読んで，ビット操作を行ってからポートに書き戻さなければならないのはいつものとおりだ。

続いてMZ-2500。先に画面モードの設定から説明する。MZ-2500には画面モードがいろいろあるので，その制御を行うためにキャラクタコントローラLSIが使われている。この石は内部にいくつかのレジスタを持っており，そこにデータを設定することで画面のモードを指定する。キャラクタコントローラはI/OポートのF4H, F5Hに割り付けられていて，F4Hにキャラクタコントローラの内部レジスタ番号をOUTしてから，F5Hにデータを送ると，そのレジスタに値がセットされる。なお，設定した値を読み出すことはできず，F4H, F5HからINすると，別の意味を持ったデータが返される。

図1　MZ-2500キャラクタコントローラ内部レジスタ0の詳細

リスト3のサブルーチンINIT2500では，キャラクタコントローラの内部レジスタのうち0～9（うち4，6は欠番）にデータをセットし，40×25行，8色，200ライン，画面0のみ表示のモードにしている。キャラクタコントローラの内部レジスタに関しては参考文献5）あたりを参考にしてもらいたいのだが，ざぁっと解説しておく。

0番のレジスタはビットごとに図1のような意味を持つ。ここでほとんど画面モードが決まる。リスト3では05Hに設定している。1，2番でテキストVRAMの表示開始アドレスを指定する。これをうまく使うと文字単位のハードウェアスクロールを行うことができる。あと，3,5,7,8番は表示範囲を指定するものだ。BASIC-M25のCONSOLE@に相当する。リストでは文献を参考に「標準的な値」に設定してある。レジスタ9はスムーズスクロール用で，ここに0～7の値をセットすることで表示位置をドット単位で指定することができる。ゲームでは関係ないので，リスト中での設定値は0だ。

ところで，上の説明のどこにも画面横幅の設定がないが，これはポートE8Hの第5ビットで行うようになっている。MZ-2000とまったく同じだ。それから，ポートF7Hの最下位ビットで8ライン/16ラインのどちらのフォントを使用するかを切り換えることができる。0なら16ライン，1なら8ラインだ。ゲームでは8ラインのPCGを使うので，1を設定している。

さて，MZ-2500では40桁モード時，テキスト2画面を持つことが可能だ。この2画面を交互に表示することで，画面書き換えの様子を隠すことができる。たとえば，画面0を表示しながら，表示されていない画面1のほうの画面に仮想画面を転送し，転送が終わったら，画面1を表示するという処理を繰り返すわけだ。E000H番地以降にテキストVRAMを割り付けたときには，E000Hからが画面0，E400Hからが画面1に相当する。画面0と画面1の開始アドレスは400Hずれている＝2進数で考えると第10ビット（上位バイトの第2ビット）が0か1かの違いなので，リスト3ではワークPAGEに書き込む画面の先頭アドレスの上位バイトを格納しておき，それをAレジスタに取り出しては，

XOR 40H

で，そのビットの0と1を反転するようにしている（注3）。そして，表示の切り換えは，さっき出てきたキャラクタコントローラのレジスタ0で設定すればよい。

で，PCGだ。PCGを表示する方法は見当がついていると思う。そう，アトリビュートVRAMをそのように設定するだけだ。PCGしか使わないのであれば，初期設定時にアトリビュートVRAMを「それもんのデータ」で埋めておけばよい。リスト3でいうと85行あたりがそうだな。こうしておけば，テキストVRAMへPCGコードを書き込むだけでPCGの表示が行える。というわけで，アトリビュートVRAMの初期設定データを直すだけで，表示処理には手を加える必要がない。残る問題はPCGの定義をする部分だろう。リスト3のサブルーチンDEFCHRを見ながら以下を読んでもらいたい。

MZ-2500でのPCG定義はメモリブロックの39Hにパターンを書き込むことで行う。メモリブロック39Hは400Hバイトごとに4つのブロックに分割されており，順にPCG0～3に対応している。8色のPCGはこのうちPCG1～3を重ね合わせることで実現される。それぞれが青・赤・緑に対応するわけだ。これらのブロックはさらに8バイトごとに区切られ，先頭からPCGコード00H～

FFHのパターンを格納するための領域となっている。「PCGnのPCGコードm」に対応するパターン格納領域オフセットアドレス（メモリブロック39Hの先頭からの相対アドレス）は，

n×400H＋m×8

で求めることができるだろう。ここまでわかればメモリブロック39Hをメインメモリに割り付けて(注4)，パターンをバシバシと書き込んでやればPCGの定義ができる。

リスト3ではラベルPCGPATで示される行以下にPCGコード80H～FFHに対応するパターンが青・赤・緑の順で並んでいるものとして，128個の8色PCGを定義している。いまのところ128個ものPCGは使っていないが，将来キャラクタの種類を増やすことなどを考えて，一応これだけの数を定義できるようにしておいた。

リスト3の230行以降に，サンプルとして用意したPCGデータとキャラクタとの対応が書いてあるから，適当にパターンを差し替えて遊んでもらいたい。

注3) E0H XOR 04H＝E4H, E4H XOR 04H＝E0H，というようにXORは任意のビットの交互反転に利用することができる。
注4) テキストVRAM（メモリブロック38H）をメモリに割り付けたときのことを思い出してほしい。同じような処理だね。

色をつける：X1/X1 turbo編

やはり画面モードの設定について先に話しておく。X1での画面設定はCRTコントローラLSI (CRTC) を制御することで行う。このCRTC内部のレジスタに値をセットすることで画面の表示状態が設定される。内部レジスタには0～17（うち10, 11と14以降はX1では使用されていない）までの番号が付いており，I/Oポートの1800H番地にレジスタ番号をOUTしてから，データをポート1801HにOUTして設定を行う。

リスト4のサブルーチンINITではCRTCの内部レジスタ0～13にデータの設定を行っている。ここで使われているOUTIという命令はLDIのOUT版で，

```
    DEC   B
    OUT   (C), (HL)
    INC   HL
```

に相当する処理を1命令で行うものだ。チェックしておかなければならないのは，Bレジスタをデクリメントしてから OUT動作が行われることで，X1のようにBCレジスタで間接的にI/Oアドレスを指定する場合には，サバを読んでBをひとつ大きくしておかないととんでもないことになる。OUTIのために，どれだけのX1ユーザーが苦杯をなめさせられたことだろう。先達の苦労を無駄にしてはいけない！

さて，CRTCに関しては参考文献4)を見てもらったほうが早いし，よくわかると思う，と逃げて，ここでは詳しい解説はしない。ただ，レジスタ12, 13でVRAMの表示開始アドレスを指定することだけは覚えておいてもらおう。レジスタ12が上位，13が下位バイトだ。表示開始アドレスを0000Hにすれば画面0，0400Hを指定すれば画面1が表示される。これは例によって，「チラツキ隠しの2画面切り換え」で利用する。このあたりに関してはMZのところの説明と，リスト4の154行以降を見てもらいたい。

PCGの表示に関しても，MZ-2500のところで話したように，アトリビュートVRAMを設定しておくだけの話なので，説明することはなにもない。地獄を見るのはPCG定義を行う部分だ。X1のPCG定義はと一ってもめんどくさい。めんどくさいと言ったらめんどくさい。説明せずに逃げたい気分なんだが，とりあえず誠意のかけらを見せて，大雑把に話してみる。が，以下の説明は十分なものではないので，でーっと読み流してから，参考文献4)で補ってもらいたい。

まずブラウン管（CRT）の表示原理を思い出してほしい。左上から順にビームを蛍光面に当てて光らせていき(走査して)，右端までいったら左に戻って，1ライン下を同じように走査する。そして，右下までいったら1画面の表示が終わったことになるので，左上に戻って，同じことを繰り返す。

次にテキストVRAMの構造を思い出してほしい。テキストVRAMには文字コードとアトリビュートが書き込まれているわけだ。メモリに単なる数字が書き込まれているだけなのに，画面には文字パターンが表示されるのはなぜかといえば，CRTCがVRAMを調べて，対応する文字フォントが格納されているCGROM上のアドレスを計算し，そこからパターンを読み出してくれているお陰だ。ここで，アトリビュートがPCGに指定されていれば，ROMの代わりにPCGパターンが格納されているRAMからパターンを取り出してくる。このCGROMないしCGRAMにCPUから直接アクセスできれば，パターンの読み出しやPCG定義が行える理屈だが，X1ではCGROM/RAMのありかはCRTCしか知らない。

ではどうやってPCGの定義を行うかと言うと，「CRTCが表示しようと思ってCGRAM上のアドレスを計算したときに，横から割り込んでそのアドレスにパターンを書き込んでしまう（この表現にはあまり自信がない)」という方法が取られている。ただし，メモリに対して読むことと書くことを同時に行うことはできない（できるように作られたメモリはある）という問題があるので，話はさらにややこしくなる。

ここでブラウン管に話を戻すと，1ライン走査して次のラインの頭に戻ろうとしているとき(水平帰線期間)と，下まで走査して上に戻るとき(垂直帰線期間)にはなにも表示されていない＝CRTCは文字パターンにアクセスしていないことがわかり，この瞬間にならPCGが定義できるのではないかと予想される。

ひとまず水平帰線期間は忘れてもらって，垂直帰線期間に話を絞る。いま垂直帰線期間に入ったとする。CRTCは文字パターンを読み出すのをしばらく休んでも構わないはずだ。ところが，律儀なCRTCはまだVRAMを読んでは文字パターンを読み出そうとする。ディスプレイに表示されるテキスト画面は最大で80×25＝2000文字なのに対してテキストVRAMは2048バイトあるが，CRTCはこの「表示されていないテキストVRAM」も調べて文字パターンのアドレスを計算し，文字パターンを読み出そうとする。この「表示はされないがCRTCがアクセスするVRAM」は画面横幅や表示開始アドレスによっても異なるが，40桁モードでページ0を表示しているときには33E8H～33FFHがそうだ。

それで，X1ではこの瞬間にだけPCGの定義ができることになっている。たぶん，この期間にはCRTCが文字パターンを格納したCGROM/RAMにアクセスしようとしてもできないような細工がしてあるのだろう(と僕は勝手に解釈している)。それでもCRTCはフォントパターンの格納されているアドレスを計算するだけするから，このアドレスを横取りして，そのメモリにPCGパターンを書き込めば，PCGの定義ができることになる。

40桁ページ0表示に限定して話をまとめると，テキストVRAMの33E8H以降を「定義したいPCGコード」で埋めて（アトリビュートもPCG表示にしておく)，垂直帰線期間を待ち，パターンを送ってやればPCGが定義される（実はこの先もまだまだひと筋縄ではいかないんだが)。

と，こんな感じの処理を行っているのがリスト4のサブルーチンDEFCHRだ。MZ-2500と同様80H～FFHまでの128個のPCGを定義している。PCG定義の本体は32行以

下の下請けサブルーチンで行っている。ここでは青8バイト・赤8バイト・緑8バイトの順に並んでいるPCGパターンデータを緑・赤・青・緑・赤……というように1バイトごとに並べ換えてから，俗に言う3倍速定義とかいうヤツでもって，1垂直帰線期間に1個の割合でPCG定義している。

＊　　　　＊

今月追加した処理は以上だ。これ以外にも細かな部分が改良・修正されているが，リストの注釈を見て納得してもらいたい。特にジャンプテーブルという手法についてのみ囲み記事で解説しておく。59ページを参照してほしい。

次回は，この連載の最終回ということで，自機の当たり判定とか，敵キャラの爆発といったやり残した部分を拾い集めて，ついでに，ジョイスティックはどーしたとか，MZ-1500のPCGが泣いてるぞとか，敵キャラが2種類だけ？　といった声にも応えてみたいと考えている。

最後になったが，リスト中の自機と敵キャラ2種類のPCGパターンは，多数(?)の応募のなかから選ばれた，鈴木哲也さんのものを使わせてもらった。ひとつも応募が来なければ，危うく真っ赤な四角の自機に真っ青な四角の敵が襲いかかるところだったが，一応格好がついてホッとしている。めでたくキャラクタパターンの入選となった鈴木さんには「その筋キーホルダー」をお送りしたい。

では，今月のリストをよく「読んで」おくことを宿題としつつ，来月の最終回へと続く。

〈参考文献〉

1)　MZ-700オーナーズマニュアル，シャープ
2)　MZ-2000オーナーズマニュアル，シャープ
3)　MZ-2500オーナーズマニュアル，シャープ
4)　祝一平：試験に出るX1，日本ソフトバンク
5)　MZ-2500テクニカルマニュアル，工学社

リスト1　MZ-700/1500用

```
  1 ;---------------------------------------------------------------
  2 ;       機種依存部      MZ-700/1500用
  3 ;
  4 ;
  5         ORG     04E00H          ;ソースがでかくなったから
  6 ;
  7 START:  CALL    INIT700         ;画面の初期化
  8         JP      MAIN            ;メインルーチンへ
  9 ;
 10 ;       画面の初期化
 11 ;
 12 INIT700:
 13         OUT     (0E3H),A        ;念のためVRAMへバンク切り換え
 14         LD      HL,0D000H       ;テキストVRAMを
 15         LD      DE,0D001H       ;スペースのキャラクタで
 16         LD      BC,0800H        ;埋める
 17         LD      (HL),SPCHR      ;
 18         LDIR                    ;
 19         LD      BC,0800H-1      ;アトリビュートVRAMを
 20         LD      (HL),40H        ;黒にセット
 21         LDIR                    ;
 22         RET
 23 ;
 24 ;       仮想VRAM->実VRAM転送
 25 ;        (VRAMはD000H以降)
 26 ;
 27 DISP700:
 28 DISP:   LD      HL,4*48+4+VRAMMODOKI
 29                                 ;HL=仮想VRAM(4,4)のアドレス
 30         LD      DE,0D000H       ;D=実VRAM(0,0)アドレス
 31         CALL    DISP00          ;テキストVRAMへの書き込み
 32         LD      HL,4*48+4+AVRAMMODOKI
 33                                 ;HL=仮想AVRAM(4,4)のアドレス
 34         LD      DE,0D800H       ;D=実AVRAM(0,0)アドレス
 35 DISP00: EXX
 36         LD      B,24            ;画面縦サイズ
 37 DISP0:  EXX
 38         LD      BC,40           ;実画面横サイズ
 39         LDIR                    ;横40文字分ブロック転送
 40         LD      A,8             ;HL=HL+A
 41         ADD     A,L             ;
 42         LD      L,A             ;
 43         JR      NC,DISP1        ;
 44         INC     H               ;
 45 DISP1:  EXX
 46         DJNZ    DISP0           ;縦サイズ分繰り返す
 47         RET
 48 ;...............................................................
 49 ;
 50 ;       キー入力
 51 ;
 52 ;
 53 INKEY700:
 54 INKEY:  LD      HL,0E001H
 55         LD      A,80H           ;A=1000_0000B
 56         LD      (0E000H),A      ;キーストローブ0
 57         LD      C,(HL)          ;C=キーデータ0
 58         ADD     A,6             ;A=1000_0110B
 59         LD      (0E000H),A      ;キーストローブ6
 60         LD      D,(HL)          ;D=キーデータ6
 61         INC     A               ;A=1000_0111B
 62         LD      (0E000H),A      ;キーストローブ7
 63         LD      E,(HL)          ;E=キーデータ7
 64         LD      A,0FFH          ;A=キー入力無し
 65         BIT     6,C             ;GRAPHキー?
 66         JR      NZ,KEY1         ;そうでなければスキップ
 67         AND     0BFH            ;bit6をリセット
 68 KEY1:   BIT     4,D             ;スペースキー?
 69         JR      NZ,KEY2         ;そうでなければスキップ
 70         AND     0DFH            ;bit5をリセット
 71 KEY2:   RR      E               ;2ビットはゴミ
 72         RR      E               ;
 73         LD      HL,CNVTBL       ;HL=変換テーブル先頭
 74         LD      B,4             ;以下のループで
 75 KEY3:   RR      E               ;  カーソルキーが
 76         JR      C,KEY4          ;  押されているかどうかを
 77         AND     (HL)            ;  順に調べて
 78 KEY4:   INC     HL              ;  対応するビットを
 79         DJNZ    KEY3            ;  リセットする
 80         RET
 81 ;
 82 ;       キー入力データ変換テーブル
 83 ;
 84 CNVTBL: DEFB    0FBH            ;カーソル左
 85         DEFB    0F7H            ;カーソル右
 86         DEFB    0FDH            ;カーソル下
 87         DEFB    0FEH            ;カーソル上
 88 ;...............................................................
 89 ;
 90 ;       背景展開処理本体
 91 ;               MZ-700では共通部のEXMAP0に代わって、
 92 ;               以下のサブルーチンを使う
 93 ;
 94 EXMAP0: LD      DE,(HOME)       ;DE=仮想VRAM上のHOME位置アドレス
 95         EXX
 96         LD      HL,(MAPPNT)     ;HL'=マップデータポインタ
 97         LD      C,11            ;C'=横方向サイズ
 98 EXMAP1: LD      B,6             ;B'=縦方向サイズ
 99 EXMAP2: LD      A,(HL)          ;A=マップコード
100         INC     HL              ;ポインタを進める
101         EXX
102         LD      L,A             ;HL=背景コード
103         LD      H,0             ;
104         ADD     HL,HL           ;HL=HLx4x4x2
105         ADD     HL,HL           ;
106         ADD     HL,HL           ;
107         ADD     HL,HL           ;
108         ADD     HL,HL           ;
109         LD      BC,BPAT0        ;BC=背景パターン先頭
110         ADD     HL,BC           ;HL=背景コードに対応する
111                                 ;  背景パターンアドレス
112         PUSH    DE              ;表示アドレスを待避
113         CALL    EXMAP7          ;仮想テキストVRAMへ書き込む
114         EX      DE,HL           ;待避した表示アドレスを復帰し
115         EX      (SP),HL         ;  同時に次の表示アドレスを待避
116         LD      BC,48*32        ;DE=DE+48*32
117         ADD     HL,BC           ;
118         EX      DE,HL           ;
119         CALL    EXMAP7          ;仮想アトリビュートVRAMへ
120         POP     DE              ;次の表示アドレスを復帰
121         EXX
122         DJNZ    EXMAP2          ;縦サイズ分繰り返す
123         EXX
124         LD      HL,-1148        ;HL=-48x6+4
125         ADD     HL,DE           ;DEが次の列を指すように
126         EX      DE,HL           ; 補正する
127         EXX
128         DEC     C               ;横サイズ分繰り返す
129         JP      NZ,EXMAP1       ;
130         RET
131 ;
132 EXMAP7:
133         LD      A,4             ;A=パターン縦サイズ
134 EXMAP8: LDI                     ;横1ライン書き込み
135         LDI                     ;
136         LDI                     ;
137         LDI                     ;
138         EX      DE,HL           ;DEが下のラインを
139         LD      BC,44           ;  指すようにする
140         ADD     HL,BC           ;
141         EX      DE,HL           ;
142         DEC     A               ;4回繰り返す
143         JP      NZ,EXMAP8       ;
144         RET
145 ;...............................................................
146 ;
147 ;       キャラクタコード
148 ;
149 MSPAT1: DEFB    05AH,04DH,043H,043H     ;自機
150         DEFB    051H,051H,061H,061H
151 CPATA:  DEFB    04EH,04DH,042H,056H     ;敵タイプA
152         DEFB    001H,021H,021H,001H
153 CPATB:  DEFB    0FEH,0FDH,0FBH,0F7H     ;敵タイプB
154         DEFB    061H,061H,061H,061H
155 ;
156 SPCHR   EQU     00H             ;空白の文字コード
157 SPATR   EQU     00H             ;空白のアトリビュート
158 MMSLCHR EQU     0E3H            ;自弾の文字コード
159 MMSLATR EQU     71H             ;自弾のアトリビュート
160 EMSLCHR EQU     47H             ;敵弾の文字コード
161 EMSLATR EQU     21H             ;敵弾のアトリビュート
162 ;...............................................................
163 ;...............................................................
164 ;
165 ;       背景キャラクタコード
166 ;
```

```
167 BPAT0:  DEFB    072H,070H,070H,070H,071H,000H,000H,000H
168         DEFB    071H,000H,000H,000H,071H,000H,000H,000H
169         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
170         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
171 BPAT1   DEFB    072H,070H,070H,070H,071H,000H,000H,000H
172         DEFB    071H,000H,05BH,000H,071H,000H,000H,000H
173         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
174         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
175 BPAT2:  DEFB    071H,000H,000H,000H,071H,000H,000H,000H
176         DEFB    072H,070H,070H,070H,071H,000H,000H,000H
177         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
178         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
179 BPAT3:  DEFB    071H,000H,05BH,000H,071H,000H,000H,000H
180         DEFB    072H,070H,070H,070H,071H,000H,000H,000H
181         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
182         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
183 BPAT4:  DEFB    071H,000H,000H,000H,071H,000H,000H,000H
184         DEFB    0F7H,0FBH,0F7H,0FBH,0FDH,0FEH,0FDH,0FEH
185         DEFB    001H,001H,001H,001H,001H,001H,001H,001H
186         DEFB    026H,026H,026H,026H,026H,026H,026H,026H
187 BPAT5:  DEFB    0F7H,0FBH,0F7H,0FBH,0FDH,0FEH,0FDH,0FEH
188         DEFB    0F7H,0FBH,0F7H,0FBH,0FDH,0FEH,0FDH,0FEH
189         DEFB    026H,026H,026H,026H,026H,026H,026H,026H
190         DEFB    026H,026H,026H,026H,026H,026H,026H,026H
191 ;
192 ;-----------------------------------------------------------------
```

リスト2 MZ-2000/2200用

```
  1 ;-----------------------------------------------------------------
  2 ;       機種依存部      MZ-2000/2200用
  3 ;
  4 ;
  5         ORG     04E00H                  ;ソースがでかくなったから
  6 ;
  7 START:  CALL    VOPEN2000               ;VRAMオープン
  8         CALL    INIT2000                ;画面の初期化
  9         CALL    MAIN                    ;メインルーチンへ
 10         CALL    VCLOSE2000              ;VRAMクローズ
 11         XOR     A                       ;背景色を
 12         OUT     (0F4H),A                ;  黒に戻す
 13         RET
 14 ;
 15 ;       VRAMオープン
 16 ;
 17 VOPEN2000:
 18         DI                              ;割り込みを禁止
 19         IN      A,(0E8H)                ;PIOポートAから入力して
 20         OR      0C0H                    ;第6,7ビットをセットして
 21         OUT     (0E8H),A                ;出力し直す
 22         RET
 23 ;
 24 ;       VRAMクローズ
 25 ;
 26 VCLOSE2000:
 27         IN      A,(0E8H)                ;PIOポートAから入力して
 28         AND     07FH                    ;第7ビットをリセットして
 29         OUT     (0E8H),A                ;出力し直す
 30         EI                              ;割り込みを許可
 31         RET
 32 ;
 33 ;       画面の初期化
 34 ;
 35 INIT2000:
 36         IN      A,(0E8H)                ;40桁モード
 37         AND     0DFH                    ;
 38         OUT     (0E8H),A                ;
 39         LD      HL,0D000H               ;テキストVRAMを
 40         LD      DE,0D001H               ;スペースのキャラクタで
 41         LD      BC,0800H-1              ;埋める
 42         LD      (HL),SPCHR              ;
 43         LDIR                            ;
 44         LD      A,1                     ;背景色を
 45         OUT     (0F4H),A                ;  青にする
 46         LD      A,7                     ;文字色を
 47         OUT     (0F5H),A                ;  白にする
 48         RET
 49 ;
 50 ;       仮想VRAM->実VRAM転送
 51 ;        (VRAMはD000H以降に割り付けられている)
 52 ;
 53 DISP2000:
 54 DISP:   LD      HL,4*48+4+VRAMMODOKI
 55                                         ;HL=仮想VRAM(4,4)のアドレス
 56         LD      DE,0D000H               ;DE=実VRAM(0,0)アドレス
 57         EXX
 58         LD      B,24                    ;画面縦サイズ
 59 DISP0:  EXX
 60         LD      BC,40                   ;実画面横サイズ
 61         LDIR                            ;横40文字分ブロック転送
 62         LD      A,8                     ;HL=HL+A
 63         ADD     A,L                     ;
 64         LD      L,A                     ;
 65         JR      NC,DISP1                ;
 66         INC     H                       ;
 67 DISP1:  EXX
 68         DJNZ    DISP0                   ;縦サイズ分繰り返す
 69         RET
 70 ;.................................................................
 71 ;
 72 ;       キー入力
 73 ;
 74 ;
 75 INKEY2000:
 76 INKEY:  LD      C,0EAH                  ;C=PIOポートBアドレス
 77         IN      A,(0E8H)                ;PIOポートAからAに入力
 78         AND     0E0H                    ;下位5ビットをリセット
 79         OR      11H                     ;A=XXX1_0001B
 80         OUT     (0E8H),A                ;キーストローブ1
 81         INC     A                       ;A=XXX1_0010B
 82         IN      E,(C)                   ;E=キーデータ1
 83         OUT     (0E8H),A                ;キーストローブ2
 84         INC     A                       ;A=XXX1_0011B
 85         IN      D,(C)                   ;D=キーデータ2
 86         OUT     (0E8H),A                ;キーストローブ3
 87         ADD     A,8                     ;A=XXX1_1011B
 88         IN      B,(C)                   ;B=キーデータ3
 89         OUT     (0E8H),A                ;キーストローブ11
 90         IN      C,(C)                   ;C=キーデータ11
 91         LD      HL,CNVTBL               ;HL=変換テーブル先頭
 92         LD      A,0FFH                  ;A=キー入力無し
 93         RR      C                       ;GRAPHキー?
 94         JR      C,KEY1                  ;そうでなければスキップ
 95         AND     0BFH                    ;bit6をリセット
 96 KEY1:   BIT     1,B                     ;スペースキー?
 97         JR      NZ,KEY2                 ;そうでなければスキップ
 98         AND     0DFH                    ;bit5をリセット
 99 KEY2:   BIT     2,E                     ;'8'?
100         JR      NZ,KEY3                 ;そうでなければスキップ
101         AND     0FEH                    ;bit0をリセット
102 KEY3:   BIT     3,E                     ;'9'?
103         JR      NZ,KEY4                 ;そうでなければスキップ
104         AND     0F6H                    ;bit0,3をリセット
105 KEY4:   LD      B,7                     ;以下のループで
106 KEY5:   RL      D                       ;  '1'~'7'のキーが
107         JR      C,KEY6                  ;  押されているかどうかを
108         AND     (HL)                    ;  順に調べて
109 KEY6:   INC     HL                      ;  対応するビットを
110         DJNZ    KEY5                    ;  リセットする
111         RET
112 ;
113 ;       キー入力データ変換テーブル
114 ;
115 CNVTBL: DEFB    0FAH                    ;'7'
116         DEFB    0F7H                    ;'6'
117         DEFB    0FFH                    ;'5'
118         DEFB    0FBH                    ;'4'
119         DEFB    0F5H                    ;'3'
120         DEFB    0FDH                    ;'2'
121         DEFB    0F9H                    ;'1'
122 ;.................................................................
123 ;
124 ;       キャラクタコード
125 ;
126 MSPAT1: DEFB    1EH,5CH,1FH,1FH         ;自機
127 CPATA:  DEFB    85H,85H,85H,85H         ;敵タイプA
128 CPATB:  DEFB    86H,86H,86H,86H         ;敵タイプB
129 ;
130 SPCHR   EQU     20H                     ;空白の文字コード
131 MMSLCHR EQU     83H                     ;自弾の文字コード
132 EMSLCHR EQU     93H                     ;敵弾の文字コード
133 ;.................................................................
134 ;
135 ;       背景キャラクタコード
136 ;
137 BPAT0:  DEFB    08EH,090H,090H,090H,09AH,03AH,03AH,03AH
138         DEFB    09AH,03AH,03AH,03AH,09AH,03AH,03AH,03AH
139 BPAT1:  DEFB    08EH,090H,090H,090H,09AH,03AH,03AH,03AH
140         DEFB    09AH,03AH,03DH,03AH,09AH,03AH,03AH,03AH
141 BPAT2:  DEFB    09AH,03AH,03AH,03AH,09AH,03AH,03AH,03AH
142         DEFB    08EH,090H,090H,090H,09AH,03AH,03AH,03AH
143 BPAT3:  DEFB    09AH,03AH,03DH,03AH,09AH,03AH,03AH,03AH
144         DEFB    08EH,090H,090H,090H,09AH,03AH,03AH,03AH
145 BPAT4:  DEFB    09AH,03AH,03AH,03AH,09AH,03AH,03AH,03AH
146         DEFB    0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH
147 BPAT5:  DEFB    0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH
148         DEFB    0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH,0DBH
149 ;
150 ;-----------------------------------------------------------------
```

リスト3 MZ-2500用

```
  1 ;-----------------------------------------------------------------
  2 ;       機種依存部      MZ-2500用
  3 ;
  4 ;
  5         ORG     04E00H                  ;ソースがでかくなったから
  6 ;
  7 START:  DI                              ;割り込みを禁止
  8         LD      A,07H                   ;論理バンク07Hをセレクト
  9         OUT     (0B4H),A                ;
 10         IN      A,(0B5H)                ;現在割り付けられている
 11         LD      (OLDMB),A               ;物理ブロックNOを待避
 12         CALL    DEFCHR                  ;PCGの定義
 13         CALL    VOPEN2500               ;VRAMを論理バンクに割り付ける
 14         CALL    INIT2500                ;画面の初期化
 15         CALL    MAIN                    ;メインルーチンへ
 16         CALL    VCLOSE2500              ;メモリマッピングを元に戻す
 17         EI                              ;割り込みを許可
 18         RET
```

```
 19 ;
 20 ;        VRAMオープン
 21 ;
 22 VOPEN2500:
 23         LD      A,07H              ;論理バンク07Hをセレクト
 24         OUT     (0B4H),A           ;
 25         LD      A,38H              ;そこにテキストVRAMを
 26         OUT     (0B5H),A           ;        割り付ける
 27         RET
 28 ;
 29 ;        VRAMクローズ
 30 ;
 31 VCLOSE2500:
 32         LD      A,07H              ;論理バンク07Hをセレクト
 33         OUT     (0B4H),A           ;
 34         LD      A,(OLDMB)          ;保存してあったMB番号を取り出し
 35         OUT     (0B5H),A           ;論理バンクに割り付ける
 36         RET
 37 ;
 38 OLDMB:  DEFS    1                  ;物理ブロックNO待避用ワーク
 39 ;
 40 ;        PCG定義
 41 ;
 42 DEFCHR: LD      A,07H              ;論理バンク07Hをセレクト
 43         OUT     (0B4H),A           ;
 44         LD      A,39H              ;そこにPCGRAMを割り付ける
 45         OUT     (0B5H),A           ;
 46         LD      HL,PCGPAT          ;HL=PCGパターン青
 47         LD      DE,0EC00H          ;DE=PCG青（80H）
 48         CALL    SETCHR             ;1色分セット
 49         LD      HL,PCGPAT+8        ;HL=PCGパターン赤
 50         LD      DE,0F400H          ;DE=PCG赤（80H）
 51         CALL    SETCHR             ;1色分セット
 52         LD      HL,PCGPAT+16       ;HL=PCG緑
 53         LD      DE,0FC00H          ;DE=PCG緑（80H）
 54 SETCHR: LD      B,64               ;64個のPCGを定義する
 55 STCHR0: PUSH    BC                 ;ループカウンタを待避する
 56         LD      BC,8               ;パターンサイズは8バイト
 57         LDIR                       ;ブロック転送
 58         LD      C,16               ;他2色のパターンを
 59         ADD     HL,BC              ;  スキップする
 60         POP     BC                 ;ループカウンタ復帰
 61         DJNZ    STCHR0             ;64回繰り返す
 62         RET
 63 ;
 64 ;        画面の初期化
 65 ;
 66 INIT2500:
 67         LD      A,1                ;200ラインモード
 68         OUT     (0F7H),A           ;
 69         IN      A,(0E8H)           ;40桁モード
 70         AND     0DFH               ;
 71         OUT     (0E8H),A           ;
 72         LD      C,0F5H             ;C=CRTCデータ出力ポート
 73         LD      HL,CRTCDAT         ;CRTC初期データ
 74         LD      B,8                ;B=ループカウンタ
 75 INILP:  LD      A,(HL)             ;A=CRTCレジスタ番号
 76         INC     HL
 77         OUT     (0F4H),A           ;CRTCレジスタセレクト
 78         OUTI                       ;データ出力
 79         JR      NZ,INILP           ;Bが0になるまで繰り返す
 80         LD      HL,0E000H          ;テキストVRAM1を
 81         LD      DE,0E001H          ;スペースのキャラクタで
 82         LD      BC,0800H           ;埋める
 83         LD      (HL),SPCHR         ;
 84         LDIR                       ;
 85         LD      BC,0800H           ;アトリビュートVRAMを
 86         LD      (HL),0FH           ;8色PCGにセット
 87         LDIR                       ;
 88         LD      BC,0800H-1         ;テキストVRAM2を
 89         LD      (HL),0             ;0で埋める
 90         LDIR                       ;
 91         RET
 92 ;
 93 ;        CRTC初期設定データ
 94 ;
 95 CRTCDAT:
 96         DEFB    0,05H              ;表示ページ0,16色モード
 97         DEFB    1,00H              ;表示開始アドレスL
 98         DEFB    2,00H              ;                H
 99         DEFB    3,26H              ;表示開始ライン
100         DEFB    5,0FEH             ;表示終了ライン
101         DEFB    7,0AH              ;表示開始桁
102         DEFB    8,5AH              ;表示終了桁
103         DEFB    9,00H              ;スムーススクロール用
104 ;
105 ;        仮想VRAM->実VRAM転送
106 ;        （VRAMはE000H以降に割り付けられている）
107 ;
108 DISP2500:
109 DISP:   LD      HL,4*48+4+VRAMMODOKI
110                                    ;HL=仮想VRAM(4,4)のアドレス
111         LD      A,(PAGE)           ;アクティブページを
112         XOR     04H                ;切り換えて
113         LD      (PAGE),A           ;ワークにセット
114         LD      D,A                ;DE=実VRAM(0,0)アドレス
115         LD      E,0                ;
116         EXX
117         LD      B,24               ;画面縦サイズ
118 DISP0:  EXX
119         LD      BC,40              ;画面横サイズ
120         LDIR                       ;横40文字分ブロック転送
121         LD      A,8                ;HL=HL+8
122         ADD     A,L                ;
123         LD      L,A                ;
124         JR      NC,DISP1           ;
125         INC     H                  ;
126 DISP1:  EXX
127         DJNZ    DISP0              ;縦サイズ分繰り返す
128         LD      A,(PAGE)           ;
129         LD      A,(CRTCDAT+1)      ;表示ページを
130         XOR     0CH                ;切り換える用意をして
131         LD      (CRTCDAT+1),A      ;ワークにセット
132         LD      C,A                ;待避
133         XOR     A                  ;表示ページを切り換える
134         OUT     (0F4H),A           ;
135         LD      A,C                ;
136         OUT     (0F5H),A           ;
137         RET
138 ;
139 PAGE:   DEFB    0E0H
140 ;...........................................................................
141 ;
142 ;        キー入力
143 ;
144 INKEY2500:
145 INKEY:  LD      C,0EAH             ;C=PIOポートBアドレス
146         IN      A,(0E8H)           ;PIOポートAからAに入力
147         AND     0E0H               ;下位5ビットをリセット
148         OR      11H                ;A=XXX1_0001B
149         OUT     (0E8H),A           ;キーストローブ1
150         INC     A                  ;A=XXX1_0010B
151         IN      E,(C)              ;E=キーデータ1
152         OUT     (0E8H),A           ;キーストローブ2
153         INC     A                  ;A=XXX1_0011B
154         IN      D,(C)              ;D=キーデータ2
155         OUT     (0E8H),A           ;キーストローブ3
156         ADD     A,7                ;A=XXX1_1010B
157         IN      B,(C)              ;B=キーデータ3
158         OUT     (0E8H),A           ;キーストローブ10
159         IN      C,(C)              ;C=キーデータ10
160         LD      HL,CNVTBL          ;HL=変換テーブル先頭
161         LD      A,0FFH             ;A=キー入力無し
162         BIT     5,C                ;ESCキー？
163         JR      NZ,KEY1            ;そうでなければスキップ
164         AND     0BFH               ;bit6をリセット
165 KEY1:   BIT     1,B                ;スペースキー？
166         JR      NZ,KEY2            ;そうでなければスキップ
167         AND     0DFH               ;bit5をリセット
168 KEY2:   BIT     2,E                ;'8'？
169         JR      NZ,KEY3            ;そうでなければスキップ
170         AND     0FEH               ;bit0をリセット
171 KEY3:   BIT     3,E                ;'9'？
172         JR      NZ,KEY4            ;そうでなければスキップ
173         AND     0F6H               ;bit0,3をリセット
174 KEY4:   LD      B,7                ;以下のループで
175 KEY5:   RL      D                  ;  '1'～'7'のキーが
176         JR      C,KEY6             ;  押されているかどうかを
177         AND     (HL)               ;  順に調べて
178 KEY6:   INC     HL                 ;  対応するビットを
179         DJNZ    KEY5               ;  リセットする
180         RET
181 ;
182 ;        キー入力データ変換テーブル
183 ;
184 CNVTBL: DEFB    0FAH               ;'7'
185         DEFB    0F7H               ;'6'
186         DEFB    0FFH               ;'5'
187         DEFB    0FBH               ;'4'
188         DEFB    0F5H               ;'3'
189         DEFB    0FDH               ;'2'
190         DEFB    0F9H               ;'1'
191 ;...........................................................................
192 ;
193 ;        キャラクタコード
194 ;
195 MSPAT1: DEFB    0A0H,0A1H,0A2H,0A3H      ;自機
196 CPATA:  DEFB    0B0H,0B1H,0B2H,0B3H      ;敵タイプA
197 CPATB:  DEFB    0B4H,0B5H,0B6H,0B7H      ;敵タイプB
198 ;
199 SPCHR   EQU     80H                ;空白の文字コード
200 MMSLCHR EQU     0A4H               ;自弾の文字コード
201 EMSLCHR EQU     0A5H               ;敵弾の文字コード
202 ;...........................................................................
203 ;
204 ;        背景キャラクタコード
205 ;
206 BPAT0:  DEFB    083H,081H,081H,081H,084H,085H,085H,085H
207         DEFB    084H,085H,085H,085H,084H,085H,085H,085H
208 BPAT1:  DEFB    083H,081H,081H,081H,084H,085H,085H,085H
209         DEFB    084H,085H,082H,085H,084H,085H,085H,085H
210 BPAT2:  DEFB    084H,085H,085H,085H,084H,085H,085H,085H
211         DEFB    083H,081H,081H,081H,084H,085H,085H,085H
212 BPAT3:  DEFB    084H,085H,082H,085H,084H,085H,085H,085H
213         DEFB    083H,081H,081H,081H,084H,085H,085H,085H
214 BPAT4:  DEFB    084H,085H,085H,085H,084H,085H,085H,085H
215         DEFB    090H,091H,090H,091H,092H,093H,092H,093H
216 BPAT5:  DEFB    090H,091H,090H,091H,092H,093H,092H,093H
217         DEFB    090H,091H,090H,091H,092H,093H,092H,093H
218 ;
219 ;---------------------------------------------------------------------------
220 ;---------------------------------------------------------------------------
221 ;
222 ;        PCGパターン
223 ;                80-8F...当たっても痛くない背景
224 ;                90-9F...当たると痛い背景
225 ;                A0-A3...自機
226 ;                A4,A5...自弾、敵弾
227 ;                A6-AF...爆発（リザーブ）
228 ;                B0-FF...敵キャラ
229 ;
230 PCGPAT: DEFB    000H,000H,000H,000H,000H,000H,000H,000H ;80H
231         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
232         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
233         DEFB    000H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;81H
234         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
235         DEFB    000H,0FFH,000H,000H,000H,000H,000H,000H
236         DEFB    0FFH,0FFH,0C3H,0FFH,0C3H,0FFH,0FFH,0FFH ;82H
237         DEFB    000H,000H,03CH,000H,03CH,000H,000H,000H
238         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
239         DEFB    000H,07FH,07FH,07FH,07FH,07FH,07FH,07FH ;83H
240         DEFB    000H,000H,000H,000H,000H,000H,000H,000H
241         DEFB    000H,07FH,040H,040H,040H,040H,040H,040H
242         DEFB    07FH,07FH,07FH,07FH,07FH,07FH,07FH,07FH ;84H
```

▶リクルートのおかげで，私と同じ名前が毎日のように新聞を賑わせている。これでこの漢字が読めるようになってくれた人はたくさん増えただろうが，印象がよくない。喜んでいいのか，悲しんでいいのか……。

江副　滋　(19)　東京都

```
243          DEFB    000H,000H,000H,000H,000H,000H,000H,000H
244          DEFB    040H,040H,040H,040H,040H,040H,040H,040H
245          DEFB    0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;85H
246          DEFB    000H,000H,000H,000H,000H,000H,000H,000H
247          DEFB    000H,000H,000H,000H,000H,000H,000H,000H
248          DEFS    24*10                                   ;86H-8FH
249          DEFB    07FH,040H,040H,040H,047H,044H,044H,044H ;90H
250          DEFB    07FH,07FH,07FH,07FH,07FH,07FH,07FH,07FH
251          DEFB    07FH,040H,040H,040H,047H,044H,044H,044H
252          DEFB    0FFH,001H,001H,001H,0F1H,001H,001H,001H ;91H
253          DEFB    0FFH,0FFH,0FFH,0FFH,0FFH,0EFH,0EFH,0EFH
254          DEFB    0FFH,001H,001H,001H,0F1H,001H,001H,001H
255          DEFB    044H,044H,040H,040H,040H,040H,07FH,000H ;92H
256          DEFB    07FH,07FH,078H,07FH,07FH,07FH,07FH,000H
257          DEFB    044H,044H,040H,040H,040H,040H,07FH,000H
258          DEFB    001H,001H,001H,001H,001H,001H,0FFH,000H ;93H
259          DEFB    0EFH,0EFH,00FH,0FFH,0FFH,0FFH,0FFH,000H
260          DEFB    001H,001H,001H,001H,001H,001H,0FFH,000H
261          DEFS    24*12                                   ;94H-9FH
262          DEFB    07FH,067H,041H,03EH,0F9H,081H,00FH,080H ;A0H
263          DEFB    000H,000H,000H,000H,01EH,0E1H,010H,0E0H
264          DEFB    058H,066H,041H,03EH,0FFH,081H,01FH,080H
265          DEFB    0FFH,0FFH,0FFH,07FH,0FFH,0E0H,00FH,0FFH ;A1H
266          DEFB    000H,000H,000H,000H,000H,0E0H,00CH,0F0H
267          DEFB    000H,000H,080H,040H,0FFH,0E0H,00FH,0FCH
268          DEFB    0FDH,0FFH,0F3H,0F7H,0FDH,0FFH,0FFH,0FFH ;A2H
269          DEFB    03FH,000H,00CH,00AH,00BH,004H,003H,000H
270          DEFB    0FDH,03FH,000H,017H,01FH,00FH,007H,003H
271          DEFB    0FFH,0FFH,0FFH,0E7H,0CBH,0FFH,0BBH,0C7H ;A3H
272          DEFB    000H,000H,000H,018H,0F4H,020H,0C4H,038H
273          DEFB    0F0H,000H,000H,0C0H,0C0H,0F0H,0A0H,080H
274          DEFB    0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;A4H
275          DEFB    000H,000H,000H,0FFH,000H,000H,000H,000H
276          DEFB    000H,000H,000H,0FFH,0FFH,000H,000H,000H
277          DEFB    0C3H,081H,00CH,004H,000H,000H,081H,0C3H ;A5H
278          DEFB    03CH,07EH,0FFH,0FFH,0FFH,0FFH,07EH,03CH
279          DEFB    000H,000H,00CH,004H,000H,000H,000H,000H
280          DEFS    10*24                                   ;A6H-AFH
281          DEFB    0FFH,0FEH,0FCH,0E3H,0DFH,0DFH,08FH,00FH ;B0H
282          DEFB    0FEH,001H,003H,00CH,000H,020H,070H,0F0H
283          DEFB    0FEH,0FDH,002H,000H,007H,00FH,067H,0E6H
284          DEFB    031H,03FH,01FH,08FH,0F7H,0F7H,0FFH,0FFH ;B1H
285          DEFB    0DFH,0C1H,0C0H,060H,000H,0C0H,060H,022H
286          DEFB    0DDH,0DFH,046H,000H,0C0H,0E0H,0F6H,06FH
287          DEFB    00EH,00EH,08FH,0DFH,0FFH,0FFH,0F3H,0FCH ;B2H
288          DEFB    0F1H,0F1H,070H,020H,000H,00DH,00FH,003H
289          DEFB    0C5H,0E4H,066H,00FH,023H,000H,00CH,003H
290          DEFB    07FH,07FH,0FFH,0FFH,0F0H,0E9H,099H,07FH ;B3H
291          DEFB    081H,081H,000H,000H,00FH,066H,0E6H,080H
292          DEFB    02FH,02FH,06FH,0E0H,000H,000H,066H,080H
293          DEFB    0FEH,0FEH,0FEH,0FFH,003H,039H,0FFH,0FFH ;B4H
294          DEFB    000H,001H,000H,070H,0FDH,0FFH,07CH,03EH
295          DEFB    001H,000H,001H,07EH,001H,039H,03CH,03EH
296          DEFB    00FH,007H,087H,0C5H,0E4H,0F4H,0FCH,086H ;B5H
297          DEFB    004H,080H,080H,0FCH,09CH,00CH,03CH,000H
298          DEFB    0F4H,078H,078H,0C6H,0E7H,037H,03FH,079H
299          DEFB    0FFH,0FEH,0FEH,0FDH,0FDH,0FBH,0F7H,0CFH ;B6H
300          DEFB    001H,000H,004H,005H,001H,001H,000H,000H
301          DEFB    001H,002H,006H,005H,00DH,01BH,037H,00FH
302          DEFB    0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;B7H
303          DEFB    09CH,08CH,000H,000H,0B0H,0F8H,0FCH,03CH
304          DEFB    0FCH,0FCH,0F0H,0E0H,0B0H,0F8H,0FCH,0BCH
305          DEFS    72*24                                   ;B8H-FFH
306 ;
307 ;-----------------------------------------------------------------
```

リスト4 X1/X1turbo用

```
  1 ;-----------------------------------------------------------------
  2 ;       機種依存部      X1/X1turbo用
  3 ;
  4 ;
  5         ORG     04E00H          ;ソースがでかくなったから
  6 ;
  7 START:  CALL    INITX1          ;画面の初期化
  8         CALL    DEFCHR          ;PCGの定義
  9         JP      MAIN            ;メインルーチンへ
 10 ;
 11 ;       PCG定義(40x25,表示ページ0専用)
 12 ;               漢字VRAMとアトリビュートVRAMは
 13 ;               初期化されていること
 14 ;
 15 DEFCHR: LD      HL,PCGPAT+16    ;HL=PCGデータ緑
 16         LD      A,80H           ;A=PCGコード
 17 DEFCH0: PUSH    HL
 18         PUSH    AF
 19         CALL    SETPCG          ;1つ定義
 20         POP     AF
 21         POP     HL
 22         LD      DE,24           ;ポインタを進めて
 23         ADD     HL,DE           ;
 24         INC     A               ;繰り返す
 25         JR      NZ,DEFCH0       ;
 26         RET
 27 ;
 28 ;       PCGを1つ定義
 29 ;               in      A=PCGコード
 30 ;                       HL=PCGパターン
 31 ;
 32 SETPCG: LD      BC,33E8H        ;見えないテキストVRAMの頭
 33         LD      E,24            ;24バイトある
 34 FILVLP: OUT     (C),A           ;定義したいPCGコードで
 35         INC     C               ;  埋めておく
 36         DEC     E               ;
 37         JR      NZ,FILVLP       ;
 38         CALL    REORD           ;PCGデータを都合良く並べ換える
 39         LD      HL,VRAMMODOKI   ;HL=並べ換えられたデータ
 40         LD      BC,1700H+100H   ;PCG緑
 41         LD      E,8             ;ループカウンタ
 42         DI                      ;割り込みを禁止して
 43         EXX                     ;垂直帰線信号の
 44         LD      BC,1A01H        ;  立ち上がりをチェック
 45 VWAIT0: IN      A,(C)           ;
 46         JP      P,VWAIT0        ;
 47 VWAIT1: IN      A,(C)           ;
 48         JP      M,VWAIT1        ;
 49         EXX                     ;
 50 STPCG0: OUTI                    ;16     ;緑定義
 51         OUTI                    ;16     ;赤定義
 52         OUTI                    ;16     ;青定義
 53         INC     B               ;4      ;OUTIで減った分を
 54         INC     B               ;4      ;  補正する
 55         INC     B               ;4      ;
 56         LD      A,10            ;7      ;時間つぶし
 57 WAITL:  DEC     A               ;40     ;
 58         JR      NZ,WAITL        ;12x9+7 ;
 59         JP      SKP             ;10     ;ダミー
 60 SKP:    NOP                     ;4      ;ダミー
 61         DEC     E               ;4      ;8回繰り返す
 62         JP      NZ,STPCG0       ;10     ;
 63         EI                              ;割り込みを許可
 64         RET
 65 ;
 66 ;       PCGデータをG0,R0,B0,G1,R1,B1,...,G7,R7,B7に
 67 ;               並べ換える
 68 ;               (変換バッファとしてVRAMMODOKIを使用)
 69 ;
 70 REORD:  LD      DE,VRAMMODOKI   ;DE=変換先
 71         LD      BC,-8           ;
 72         EXX
 73         LD      L,8             ;ループカウンタ
 74 REOD0:  EXX
 75         PUSH    HL              ;データポインタ待避
 76         LD      A,3             ;ループカウンタ
 77 REOD1:  LDI                     ;1バイト転送
 78                                 ;このときBC=-9
 79         ADD     HL,BC           ;HL=HL-9
 80         INC     C               ;BC=-8
 81         DEC     A               ;3回繰り返す
 82         JR      NZ,REOD1        ;
 83         POP     HL              ;データポインタ復帰
 84         INC     HL              ;データポインタを進め
 85         EXX
 86         DEC     L               ;8回繰り返す
 87         JR      NZ,REOD0        ;
 88         RET
 89 ;
 90 ;       画面の初期化
 91 ;
 92 INITX1:
 93 ;       ................turboでは以下の行を復活する
 94 ;       XOR     A
 95 ;       LD      BC,1FD0H
 96 ;       OUT     (C),A
 97 ;       ................ここまでがturbo用
 98         LD      A,30H           ;A=テキストVRAM先頭アドレスH
 99         LD      (PAGE),A        ;アクティブページ設定
100         LD      BC,1A02H        ;BC=8255_2
101         LD      A,-1            ;第6ビットが1なら40桁モード
102         OUT     (C),A           ;40桁モードへ
103         LD      HL,CRTCDAT      ;HL=CRTC設定データポインタ
104         XOR     A               ;A=CRTCレジスタ0
105         LD      BC,1800H        ;BC=CRTCレジスタ選択ポート
106         LD      E,14            ;E=ループカウンタ
107 ILOOP0: OUT     (C),A           ;CRTCレジスタセレクト
108         INC     B               ;OUTI用に1増やしておく
109         INC     C               ;BC=CRTCデータポート+100H
110         OUTI                    ;PORT(BC)=(HL)+
111         DEC     C               ;BC=CRTCレジスタ選択ポート
112         INC     A               ;CTRCレジスタ番号を増やす
113         DEC     E               ;E=0まで繰り返す
114         JR      NZ,ILOOP0       ;
115         LD      BC,2800H        ;BC=AVRAM先頭
116         LD      DE,8027H        ;D=空白キャラクタコード
117                                 ;E=アトリビュート(PCG)
118         LD      HL,0800H        ;HL=ループカウンタ
119 ILOOP1: OUT     (C),E           ;アトリビュートVRAMを
120         INC     BC              ;  すべてPCGで
121         DEC     HL              ;  埋める
122         LD      A,H             ;
123         OR      L               ;
124         JR      NZ,ILOOP1       ;
125         LD      H,08H           ;HL=ループカウンタ
126 ILOOP2: OUT     (C),D           ;テキストVRAMを
127         INC     BC              ;  空白のキャラクタで
128         DEC     HL              ;  埋める
129         LD      A,H             ;
130         OR      L               ;
131         JR      NZ,ILOOP2       ;
132 ;       ................turboでは以下の行を復活する
133 ;       LD      H,08H           ;HL=ループカウンタ
134 ;       LD      E,00H           ;
135 ;ILOOP3:OUT     (C),E           ;漢字VRAMを
136 ;       INC     BC              ;  0で埋める
137 ;       DEC     HL              ;
138 ;       LD      A,H             ;
139 ;       OR      L               ;
140 ;       JR      NZ,ILOOP3       ;
141 ;       ................ここまでがturbo用
142         RET
143 ;
144 ;       CRTC初期設定データ
```

▶株の譲与リストを捜し出す，究極のアドベンチャーゲーム「リクルートへの道」。またはマスコミに見つからずにいかに多くの政治献金を集めるかという「政治家への道」。このようなゲームを誰か作ってくれませんかね。　長田　純也　(19)　神奈川県

```
145 ;               200ライン,40桁x25行
146 ;
147 CRTCDAT:
148         DEFB    55,40,45,52,31,2,25,28
149         DEFB    0,7,0,0,0,0
150 ;
151 ;       仮想VRAM->実VRAM転送
152 ;        (VRAMはI/Oポートアドレス3000H以降)
153 ;
154 DISPX1:
155 DISP:   LD      HL,4*48+4+VRAMMODOKI
156                                 ;HL=仮想VRAM(4,4)のアドレス
157         LD      A,(PAGE)        ;アクティブページを
158         XOR     04H             ;切り換えて
159         LD      (PAGE),A        ;ワークにセット
160         LD      B,A             ;BC=実VRAM(0,0)アドレス
161         LD      C,00H           ;
162         LD      D,24            ;画面縦サイズ
163 DISP0:  LD      E,40            ;実画面横サイズ
164 DISP1:  LD      A,(HL)          ;横40文字分を転送
165         INC     HL              ;
166         OUT     (C),A           ;
167         INC     BC              ;
168         DEC     E               ;
169         JR      NZ,DISP1        ;
170         LD      A,8             ;HL=HL+8
171         ADD     A,L             ;
172         LD      L,A             ;
173         JR      NC,DISP2        ;
174         INC     H               ;
175 DISP2:  DEC     D               ;縦サイズ分繰り返す
176         JR      NZ,DISP0        ;
177         LD      BC,1800H        ;CRTCポートアドレス
178         LD      A,12            ;CRTCレジスタ12をセレクト
179         OUT     (C),A           ;
180         INC     C               ;BC=1801H
181         LD      A,(PAGE)        ;上位5ビットは何でもよい
182         OUT     (C),A           ;表示ページを切り換える
183         RET
184 ;
185 PAGE:   DEFS    1
186 ;.........................................................
187 ;
188 ;       キー入力
189 ;
190 ;       ................turboでは以下の行を削除する
191 INKEYX1:
192 INKEY:  EI                      ;割り込みを許可しておく
193         LD      E,0E6H          ;キーデータ読みだしコマンド
194         CALL    SEND            ;サブCPUへ送る
195         CALL    SWAIT           ;待つ
196         DI                      ;ここで割り込みを禁止する
197         CALL    RECV            ;サブCPUから1バイト受け取る
198                                 ;けど、要らないから捨てる
199         CALL    RECV            ;サブCPUから1バイト受け取る
200         EI                      ;割り込みを許可する
201         JR      Z,NOKEY         ;A=0ならキー入力無し
202         SUB     1BH             ;ASCIIコードがESC以下なら
203         JR      C,NOKEY         ;  無効
204         CP      '9'-1BH+1       ;ASCIIコードが9以上なら
205         JR      NC,NOKEY        ;  無効
206         LD      HL,CNVTBL       ;HL=変換テーブル先頭
207         LD      C,A             ;ASCIIコードに対応する
208         LD      B,0             ;  ジョイスティック型データを
209         ADD     HL,BC           ;  テーブルから
210         LD      A,(HL)          ;  取り出す
211         RET
212 NOKEY:  LD      A,0FFH
213         RET
214 ;
215 ;       キー入力データ変換テーブル
216 ;
217 CNVTBL: DEFB    0BFH                    ; 1B (ESC)
218         DEFB    0FFH,0FFH,0FFH,0FFH     ; 1C - 1F
219         DEFB    0DFH                    ; 20 (SPACE)
220         DEFB    0FFH,0FFH,0FFH,0FFH     ; 21 - 24
221         DEFB    0FFH,0FFH,0FFH,0FFH     ; 25 - 28
222         DEFB    0FFH,0FFH,0FFH,0FFH     ; 29 - 2C
223         DEFB    0FFH,0FFH,0FFH,0FFH     ; 2D - 30
224         DEFB    0F9H                    ; 31 ('1')      1111 1001
225         DEFB    0FDH                    ; 32 ('2')      1111 1101
226         DEFB    0F5H                    ; 33 ('3')      1111 0101
227         DEFB    0FBH                    ; 34 ('4')      1111 1011
228         DEFB    0FFH                    ; 35 ('5')      1111 1111
229         DEFB    0F7H                    ; 36 ('6')      1111 0111
230         DEFB    0FAH                    ; 37 ('7')      1111 1010
231         DEFB    0FEH                    ; 38 ('8')      1111 1110
232         DEFB    0F6H                    ; 39 ('9')      1111 0110
233 ;       ................ここまでがX1用
234 ;       ................turboでは以下の行を復活する
235 ;INKEYturbo:
236 ;INKEY: EI                      ;割り込みを許可しておく
237 ;       LD      E,0E3H          ;ゲームキー読み出しコマンド
238 ;       CALL    SEND            ;サブCPUへ送る
239 ;       CALL    SWAIT           ;待つ
240 ;       DI                      ;ここで割り込みを禁止する
241 ;       CALL    RECV            ;サブCPUから1バイト受け取る
242 ;                               ;けど、要らないから捨てる
243 ;       CALL    RECV            ;サブCPUから2バイト目を受け取る
244 ;       LD      E,A             ;それをEに取っておく
245 ;       CALL    RECV            ;サブCPUから3バイト目を受け取る
246 ;       LD      D,A             ;それをDに取っておく
247 ;       EI                      ;割り込みを許可する
248 ;       LD      HL,CNVTBL       ;HL=変換テーブル先頭
249 ;       LD      B,8             ;B=ループカウンタ
250 ;       LD      A,0FFH          ;A=キー入力無し
251 ;INKYLP: RR     E               ;以下のループで
252 ;       JR      NC,KEY1         ;  '1'～'9'のキーが
253 ;       AND     (HL)            ;  押されているかどうかを
254 ;KEY1:  INC     HL              ;  順に調べて
255 ;       DJNZ    INKYLP          ;  対応ビットをリセットする
256 ;       BIT     1,D             ;スペースキー?
257 ;       JR      Z,KEY2          ;そうでなければスキップ
258 ;       AND     0DFH            ;bit5をリセット
259 ;KEY2:  RL      D               ;ESCキー?
260 ;       RET     NC              ;そうでなければリターン
261 ;       AND     0BFH            ;bit6をリセットする
262 ;       RET
263 ;
264 ;CNVTBL: DEFB   0F5H            ;'3'
265 ;       DEFB    0F7H            ;'6'
266 ;       DEFB    0F6H            ;'9'
267 ;       DEFB    0FDH            ;'2'
268 ;       DEFB    0FEH            ;'8'
269 ;       DEFB    0F9H            ;'1'
270 ;       DEFB    0FBH            ;'4'
271 ;       DEFB    0FAH            ;'7'
272 ;       ................ここまでがturbo用
273 ;
274 ;       サブCPUへ1バイト送る
275 ;
276 SEND:   CALL    SWAIT
277         LD      BC,1900H
278         OUT     (C),E
279         RET
280 ;
281 ;       サブCPUから1バイト受け取る
282 ;
283 RECV:   CALL    RWAIT
284         LD      BC,1900H
285         IN      A,(C)
286         RET
287 ;
288 ;       サブCPUへデータが送れるように
289 ;                       なるまで待つ
290 ;
291 SWAIT:  LD      BC,1A01H
292 SWAIT0: IN      A,(C)
293         RL      A
294         JP      M,SWAIT
295         RET
296 ;
297 ;       サブCPUからデータを読み込めるように
298 ;                       なるまで待つ
299 ;
300 RWAIT:  LD      BC,1A01H
301 RWAIT0: IN      A,(C)
302         AND     20H
303         JR      NZ,RWAIT0
304         RET
```

リスト5 共通部

```
 1 ;---------------------------------------------------------
 2 ;       共通部
 3 ;
 4 MMSLMAX EQU     4               ;自弾の最大数
 5 CHRMAX  EQU     32              ;敵の最大数
 6 EMSLMAX EQU     16              ;敵弾の最大数
 7 MMSLCNT EQU     3               ;リピートカウンタ初期値
 8 ;
 9 ;       自弾バッファのフィールドオフセット
10 ;
11 MSLX    EQU     0               ;X座標(44以上...DEAD)
12 MSLY    EQU     1               ;Y座標
13 MSLADRL EQU     2               ;表示アドレス   Low
14 MSLADRH EQU     3               ;               High
15 MBUFSIZ EQU     4               ;バッファサイズは4バイト
16 ;
17 ;       敵キャラバッファのフィールドオフセット
18 ;
19 TYPE    EQU     0               ;種類(-1...DEAD)
20 X       EQU     1               ;X座標
21 Y       EQU     2               ;Y座標
22 ADRL    EQU     3               ;表示アドレス
23 ADRH    EQU     4               ;
24 PATL    EQU     5               ;パターンアドレス
25 PATH    EQU     6               ;
26 WORK1   EQU     7               ;汎用ワーク1
27 WORK2   EQU     8               ;          2
28 WORK3   EQU     9               ;          3
29 BUFSIZ  EQU     10              ;バッファサイズは10バイト
30 ;
31 ;       敵弾バッファのフィールドオフセット
32 ;
33 EMX     EQU     0               ;X座標
34 EMY     EQU     1               ;Y座標
35 EMADRL  EQU     2
36 EMADRH  EQU     3
37 EMP     EQU     4               ;確率?
38 EMDP    EQU     5               ;確率の加値
39 EMDX0   EQU     6               ;X座標変位0
40 EMDY0   EQU     7               ;Y座標変位0
41 EMDX1   EQU     8               ;X座標変位1
42 EMDY1   EQU     9               ;Y座標変位1
```

▶この前，テレビのCMを見て絶句してしまった。「下松健康パークトロン湯センター」だって。いったいどんな湯なんだろう？　秋友　謙二　(15)　山口県

```
43 EMDA0L  EQU      10              ;表示アドレス変位0
44 EMDA0H  EQU      11              ;
45 EMDA1L  EQU      12              ;表示アドレス変位1
46 EMDA1H  EQU      13              ;
47 EMBFSIZ EQU      14              ;バッファサイズは12バイト
48 ;
49 DEAD    EQU      -1              ;キャラ/弾が死んでいる印
50 ;..........................................................
51 ;
52 ;        メインルーチン
53 ;
54 MAIN:   LD       HL,IVNTDAT      ;イベントデータポインタの
55         LD       (IVNTP),HL      ;      初期化
56         LD       HL,EMSLCNT      ;敵弾移動カウンタの
57         LD       (HL),0AAH       ;        初期化
58         LD       A,88H           ;
59         LD       (IVNTC),A       ;イベントカウンタの初期化
60         LD       (MAPCNT1),A     ;マップカウンタ1の初期化
61         LD       (MAPCNT2),A     ;マップカウンタ2の初期化
62         LD       HL,MAPDAT       ;マップデータポインタの初期化
63         LD       (MAPPNT),HL     ;
64         LD       HL,4*48+4+VRAMMODOKI
65         LD       (HOME),HL       ;仮想VRAM上のHOME位置初期化
66         CALL     INITMS          ;自機ワークの初期化
67         CALL     INITMMSL        ;自弾ワークの初期化
68         CALL     INITWK          ;敵ワークの初期化
69         CALL     INITEMSL        ;敵弾ワークの初期化
70 LOOP:   CALL     IVNT            ;イベント処理
71         RET      C               ;CY=1であれば全面クリア
72         CALL     EXTMAP          ;背景を仮想VRAMに展開
73         CALL     MOVEMS          ;自機の移動
74         CALL     MOVEMMSL        ;自弾の移動・表示
75         CALL     MOVETEKI        ;敵の移動・表示
76         CALL     MOVEEMSL        ;敵弾の移動・表示
77         CALL     PUTMS           ;自機の表示
78         CALL     DISP            ;仮想VRAMを表示
79         CALL     PAUSE           ;一時停止
80         JR       LOOP
81 ;..........................................................
82 ;
83 ;        自機データバッファの初期化
84 ;
85 INITMS: LD       HL,MSINID       ;HL=初期データ
86         LD       DE,MSBUF        ;DE=バッファ先頭
87         LD       BC,BUFSIZ       ;BC=バッファサイズ
88         LDIR                     ;ブロック転送
89         RET
90 ;
91 ;        自機初期データ
92 ;
93 MSINID:
94         DEFB     0,8,15          ;ノーマル,X=8,Y=15
95         DEFW     15*48+8+VRAMMODOKI
96         DEFW     MSPAT1          ;パターンもノーマル
97         DEFB     0,0,0           ;ダミー
98 ;
99 ;        自機の移動処理
100 ;
101 MOVEMS: CALL    INKEY           ;キー入力データを得る
102         LD      DE,(MSX)        ;E=X座標, D=Y座標
103         LD      HL,(MSADR)      ;HL=表示アドレス
104         LD      BC,48           ;BC=仮想VRAM横サイズ
105         RRA                     ;下へ?
106         JR      C,MSMV1         ;そうでなければスキップ
107         DEC     D               ;Y=Y-1
108         SBC     HL,BC           ;ADR=ADR-48
109 MSMV1:  RRA                     ;上へ?
110         JR      C,MSMV2         ;そうでなければスキップ
111         INC     D               ;Y=Y+1
112         ADD     HL,BC           ;ADR=ADR+48
113 MSMV2:  RRA                     ;左へ?
114         JR      C,MSMV3         ;そうでなければスキップ
115         DEC     E               ;X=X-1
116         DEC     HL              ;ADR=ADR-1
117 MSMV3:  RRA                     ;右へ?
118         JR      C,MSMV4         ;そうでなければスキップ
119         INC     E               ;X=X+1
120         INC     HL              ;ADR=ADR+1
121 MSMV4:  EX      AF,AF'          ;A'=キーデータ
122         LD      A,D             ;A=Y座標
123         CP      4               ;Y座標は最小値以下か?
124         JR      NC,MSMV5        ;そうでなければスキップ
125         INC     D               ;Y座標補正
126         ADD     HL,BC           ;表示アドレス補正
127         JR      MSMV6
128 MSMV5:  CP      27              ;Y座標は最大値以上か?
129         JR      C,MSMV6         ;そうでなければスキップ
130         DEC     D               ;Y座標補正
131         SBC     HL,BC           ;表示アドレス補正
132 MSMV6:  LD      A,E             ;A=X座標
133         CP      5               ;X座標は最小値以下か?
134         JR      NC,MSMV7        ;そうでなければスキップ
135         INC     E               ;X座標補正
136         INC     HL              ;表示アドレス補正
137         JR      MSMV8
138 MSMV7:  CP      42              ;X座標は最大値以上か?
139         JR      C,MSMV8         ;そうでなければスキップ
140         DEC     E               ;X座標補正
141         DEC     HL              ;表示アドレス補正
142 MSMV8:  EX      AF,AF'          ;A=キーデータの残り
143         LD      (MSX),DE        ;XY座標更新
144         LD      (MSADR),HL      ;表示アドレス更新
145         RRA                     ;1ビットはゴミ
146         RRA                     ;トリガー1が押されたか?
147         JR      C,TRGOFF        ;そうでなければ分岐
148         LD      A,(MSWK1)       ;A=直前のトリガーの状態
149         OR      A               ;前回はトリガーが押されていたか?
150         JR      Z,FIRE          ;前回OFFだったなら弾発射処理へ
151         LD      A,(MSWK2)       ;A=トリガーのカウンタ
152         DEC     A               ;カウンタを減らして
153         LD      (MSWK2),A       ;カウンタ更新
154         JR      NZ,FREND        ;カウンタが0でなければ射撃不可
155 FIRE:   CALL    MMSLALC         ;自弾ワークの空きを探す
156         JR      C,TRGOFF        ;空きが無ければ次回に送る
157         PUSH    HL              ;HL=自弾バッファ空き領域
158         POP     IX              ;IX=HL
159         LD      DE,(MSX)        ;DE=自機XY座標
160         INC     E               ;X=X+1,Y=Y+1(自機の右下)
161         INC     D               ;
162         LD      (IX+MSLX),E     ;自弾のX座標をセット
163         LD      (IX+MSLY),D     ;自弾のY座標をセット
164         LD      HL,(MSADR)      ;DE=自機表示アドレス
165         LD      DE,49           ;
166         ADD     HL,DE           ;ADR=ADR+49(自機の右下)
167         LD      (IX+MSLADRL),L  ;自弾の表示アドレスをセット
168         LD      (IX+MSLADRH),H  ;
169         LD      HL,MSWK1        ;今回、弾を撃ったという
170         LD      (HL),-1         ;         印をセット
171         INC     HL              ;HL=MSWK2
172         LD      (HL),MMSLCNT    ;リピートカウンタをリセット
173 FREND:  RET
174 TRGOFF: LD      HL,MSWK1        ;今回、弾を撃たなかったという
175         LD      (HL),0          ;          印をセット
176 MSMV9:  RET
177 ;
178 ;        自機の仮想VRAMへの表示
179 ;
180 PUTMS:  LD      IX,MSBUF        ;IX=自機データバッファ先頭
181         JP      PUT             ;表示ルーチンへ
182 ;..........................................................
183 ;
184 ;        自弾バッファの初期化
185 ;
186 INITMMSL:
187         LD      HL,MMSLBUF      ;HL=バッファ先頭
188         LD      B,MMSLMAX       ;B=自弾の最大数
189 INIMMSL:LD      (HL),DEAD       ;MMSLBUF.X=DEAD
190         INC     HL              ;ポインタを進める
191         INC     HL              ;
192         INC     HL              ;
193         INC     HL              ;
194         DJNZ    INIMMSL         ;自弾の最大数分繰り返す
195         RET
196 ;
197 ;        自弾バッファの空きエリアを探す
198 ;                out     CY=1 ... 空き無し
199 ;                        CY=0 ... HL=空きエリアへのポインタ
200 ;
201 MMSLALC:
202         LD      HL,MMSLBUF      ;HL=バッファ先頭
203         LD      B,MMSLMAX       ;B=自弾の最大数
204 MMALCL: LD      A,(HL)          ;A=自弾X座標
205         CP      44              ;X座標は44以上か?
206         RET     NC              ;そうなら空き領域
207         INC     HL              ;ポインタを進める
208         INC     HL              ;
209         INC     HL              ;
210         INC     HL              ;
211         DJNZ    MMALCL          ;バッファの終わりまで繰り返す
212         SCF                     ;空き領域が見つからなかった
213         RET
214 ;
215 ;        自弾移動処理
216 ;
217 MOVEMMSL:
218         LD      IX,MMSLBUF      ;IX=自弾データバッファ先頭
219         LD      B,MMSLMAX       ;B=自弾最大数
220 MVMMSL: LD      A,(IX+MSLX)     ;A=MMSLBUF.MSLX
221         CP      44              ;場外?
222         JR      NC,MVMMSL1      ;そうであれば死んでいる
223         INC     (IX)            ;X座標を2進める
224         INC     (IX)            ;
225         CALL    MMSLCHK         ;敵との当たり判定
226         JR      C,MVMMSL1       ;CY=1なら命中
227         LD      L,(IX+MSLADRL)  ;HL=表示アドレス
228         LD      H,(IX+MSLADRH)  ;
229         INC     HL              ;表示アドレスを2進める
230         INC     HL              ;
231         LD      (IX+MSLADRL),L  ;表示アドレス更新
232         LD      (IX+MSLADRH),H  ;
233         LD      (HL),MMSLCHR    ;仮想VRAMへ表示
234 ;       ...............MZ-700では以下の行を復活させる
235 ;       LD      DE,48*32        ;自弾の
236 ;       ADD     HL,DE           ;  アトリビュートを
237 ;       LD      (HL),MMSLATR    ;  書き込む
238 ;       ...............ここまでがMZ-700専用追加部分
239 MVMMSL1:INC     IX              ;ポインタを進める
240         INC     IX              ;
241         INC     IX              ;
242         INC     IX              ;
243         DJNZ    MVMMSL          ;自弾の最大数分繰り返す
244         RET
245 ;
246 ;        自弾と敵の当たり判定
247 ;
248 MMSLCHK:EXX                     ;表レジスタは保存
249         LD      IY,CHRBUF       ;IY=敵キャラデータバッファ先頭
250         LD      DE,BUFSIZ       ;DE=敵キャラバッファサイズ/1機
251         LD      B,CHRMAX        ;B=敵最大数
252 MMSLCK: LD      A,(IY+TYPE)     ;A=敵タイプ
253         INC     A               ;死んでいるか?
254         JR      Z,NOTHIT        ;そうならスキップ
255         LD      A,(IX+MSLX)     ;A=自弾X座標
256         SUB     (IY+X)          ;それから敵のX座標を引いてみて
257         CP      4               ;当たる範囲か?
258         JR      NC,NOTHIT       ;範囲外ならスキップ
259         LD      A,(IX+MSLY)     ;A=自弾Y座標
260         SUB     (IY+Y)          ;それから敵のY座標を引いてみて
261         INC     A               ;当たる範囲はちょっと広く取った!
262         CP      4               ;当たる範囲か?
263         JR      NC,NOTHIT       ;範囲外ならスキップ
264 HIT:    LD      (IX+MSLX),DEAD  ;自弾を殺す
265         LD      (IY+TYPE),DEAD  ;敵も殺す
266         SCF                     ;自弾は死んだ
```

▶我が家のお風呂は、銭湯でもないのに薪で沸かしています。だからX68000ください。文明開化の光を僕に与えてほしい。　白井　英治　(19)　京都府

```
267          EXX                         ;表レジスタ復帰
268          RET
269 NOTHIT:  ADD     IY,DE               ;ポインタを進めて
270          DJNZ    MMSLCK              ;敵の最大数分繰り返す
271          OR      A                   ;自弾は生き残った
272          EXX                         ;表レジスタ復帰
273          RET
274 ;..................................................................
275 ;
276 ;        マップデータを仮想VRAMに展開
277 ;
278 EXTMAP:  LD      HL,MAPCNT1          ;カウンタ1を進める
279          RRC     (HL)                ;
280          JP      NC,EXMAP0           ;NCなら今回はスクロールしない
281          INC     HL                  ;HL=MAPCNT1
282          RRC     (HL)                ;カウンタ2を進める
283          JR      NC,EXTMP            ;NCならスキップ
284          LD      HL,(MAPPNT)         ;マップデータポインタを
285          LD      DE,6                ;  4キャラ分
286          ADD     HL,DE               ;  進めて
287          LD      (MAPPNT),HL         ;  更新する
288          LD      HL,4*48+4+VRAMMODOKI
289          LD      (HOME),HL           ;仮想VRAM上のHOME位置をリセット
290 EXTMP:   CALL    EXMAP0              ;背景展開本体を呼び出す
291          LD      HL,(HOME)           ;仮想VRAM上のHOME位置を
292          DEC     HL                  ;  1キャラ分ずらして
293          LD      (HOME),HL           ;  更新する
294          RET
295 ;
296 ;        背景展開処理本体
297 ;
298 ;        ................MZ-700では以下の行を削除する
299 EXMAP0:  LD      DE,(HOME)           ;DE=仮想VRAM上のHOME位置アドレス
300          EXX
301          LD      HL,(MAPPNT)         ;HL'=マップデータポインタ
302          LD      C,11                ;C'=横方向サイズ
303 EXMAP1:  LD      B,6                 ;B'=縦方向サイズ
304 EXMAP2:  LD      A,(HL)              ;A=マップコード
305          INC     HL                  ;ポインタを進める
306          EXX
307          LD      L,A                 ;HL=背景コード
308          LD      H,0                 ;
309          ADD     HL,HL               ;HL=HLx4x4
310          ADD     HL,HL               ;
311          ADD     HL,HL               ;
312          ADD     HL,HL               ;
313          LD      BC,BPAT0            ;BC=背景パターン先頭
314          ADD     HL,BC               ;HL=背景コードに対応する
315                                      ;  背景パターンアドレス
316          LD      A,4                 ;A=パターン縦サイズ
317 EXMAP3:  LDI                         ;横1ライン書き込み
318          LDI                         ;
319          LDI                         ;
320          LDI                         ;
321          EX      DE,HL               ;DEが下のラインを
322          LD      BC,44               ;  指すようにする
323          ADD     HL,BC               ;
324          EX      DE,HL               ;
325          DEC     A                   ;4回繰り返す
326          JP      NZ,EXMAP3           ;
327          EXX
328          DJNZ    EXMAP2              ;縦サイズ分繰り返す
329          EXX
330          LD      HL,-1148            ;HL=-48x6+4
331          ADD     HL,DE               ;DEが次の列を指すように
332          EX      DE,HL               ;  補正する
333          EXX
334          DEC     C                   ;横サイズ分繰り返す
335          JP      NZ,EXMAP1
336          RET
337 ;        ................MZ-700ではここまでを削除する
338 ;
339 MAPCNT1:DEFS     1                   ;1キャラスクロールカウンタ
340 MAPCNT2:DEFS     1                   ;4キャラスクロールカウンタ
341 MAPPNT:  DEFS    2                   ;現在のマップデータへのポインタ
342 HOME:    DEFS    2                   ;仮想VRAM上のHOME位置アドレス
343 ;..................................................................
344 ;
345 ;        イベント処理
346 ;
347 IVNT:    LD      HL,IVNTC            ;(HL)=カウンタ
348          RRC     (HL)                ;カウンタを進める
349          RET     NC                  ;NCであればイベント無し
350          LD      HL,(IVNTP)          ;HL=イベントデータポインタ
351          LD      A,(HL)              ;A=イベントコード
352          INC     HL                  ;イベントデータポインタ++
353          LD      (IVNTP),HL          ;
354          OR      A                   ;無効イベントか?
355          RET     Z                   ;そうであればRET
356          JP      P,IVNT1             ;80H以下であれば敵登場イベント
357          CP      -1                  ;終了イベントか?
358          JR      NZ,IVNT0            ;そうでなければその他のイベント
359          SCF                         ;おしまいの印
360          RET
361 IVNT0:   SUB     80H                 ;80H以上のイベント処理へ
362          ADD     A,A                 ;  ジャンプテーブルを用いて
363          LD      E,A                 ;  分岐する
364          LD      D,0                 ;
365          LD      HL,IJMPTBL          ;
366          ADD     HL,DE               ;
367          LD      E,(HL)              ;
368          INC     HL                  ;
369          LD      D,(HL)              ;
370          EX      DE,HL               ;
371          JP      (HL)                ;
372 ;
373 IVNTP:   DEFS    2                   ;イベントデータポインタ
374 IVNTC:   DEFB    0                   ;イベントカウンタ
375 ;
376 ;        イベントジャンプテーブル
377 ;
378 IJMPTBL:
379          DEFW    IVNT80,DMY,DMY,DMY       ;80-83
380          DEFW    DMY,DMY,DMY,DMY          ;84-87
381          DEFW    DMY,DMY,DMY,DMY          ;88-8B
382          DEFW    DMY,DMY,DMY,DMY          ;8C-8F
383 ;
384 ;        イベントno.80H
385 ;        マップデータを繰り返す
386 ;
387 IVNT80:
388          LD      A,88H               ;
389          LD      (MAPCNT1),A         ;マップカウンタ1の初期化
390          LD      (MAPCNT2),A         ;マップカウンタ2の初期化
391          LD      HL,MAPDAT           ;マップデータポインタの初期化
392          LD      (MAPPNT),HL         ;
393 DMY:     OR      A
394          RET
395 ;
396 ;        敵登場イベント
397 ;
398 IVNT1:   LD      DE,BUFSIZ           ;DE=バッファサイズ/1機
399          LD      HL,ADAT1            ;HL=イベント1初期データ
400          LD      B,5                 ;今のところ敵登場イベントは5種類
401 IVNTL:   DEC     A                   ;力まかせに
402          JR      Z,APP               ;  HL=DEx(A-1)を
403          ADD     HL,DE               ;  計算する
404          DJNZ    IVNTL               ;
405          SCF
406          RET
407 APP:     CALL    WALLOC              ;バッファの空きを探す
408          JR      C,APP0              ;見つからなければ登場しない
409          LD      BC,BUFSIZ           ;BC=敵キャラバッファサイズ/1機
410          LDIR                        ;ごっそりブロック転送
411 APP0:    OR      A                   ;CY=0
412          RET
413 ;
414 ;        敵キャラバッファの空きを探す
415 ;        out     CY=1 ... 空き無し
416 ;                CY=0 ... DE=空きエリアへのポインタ
417 ;
418 WALLOC:  PUSH    HL                  ;HLを待避
419          LD      HL,CHRBUF           ;HL=敵キャラバッファ先頭
420          LD      DE,BUFSIZ           ;   DE=バッファサイズ/1機
421          LD      B,CHRMAX            ;B=敵最大数
422 WALCLP:  LD      A,(HL)              ;A=CHRBUF.TYPE
423          INC     A                   ;A=DEAD?
424          JR      Z,WRETN             ;そうであれば空きエリア
425          ADD     HL,DE               ;ポインタを進めて
426          DJNZ    WALCLP              ;バッファの終わりまで繰り返す
427          SCF                         ;空き領域が見つからなかった
428 WRETN:   EX      DE,HL               ;CY=0であればDE=空き領域
429          POP     HL                  ;HLを復帰
430          RET
431 ;
432 ;        敵登場イベント初期データ
433 ;
434 ADAT1:   DEFB    0,44,10             ;イベント1
435          DEFW    10*48+44+VRAMMODOKI
436          DEFW    CPATA
437          DEFB    0,0,0
438 ;
439 ADAT2:   DEFB    0,44,20             ;イベント2
440          DEFW    20*48+44+VRAMMODOKI
441          DEFW    CPATA
442          DEFB    0,0,0
443 ;
444 ADAT3:   DEFB    1,44,10             ;イベント3
445          DEFW    10*48+44+VRAMMODOKI
446          DEFW    CPATB
447          DEFW    MOVDATB
448          DEFB    0
449 ;
450 ADAT4:   DEFB    1,44,15             ;イベント4
451          DEFW    15*48+44+VRAMMODOKI
452          DEFW    CPATB
453          DEFW    MOVDATB
454          DEFB    0
455 ;
456 ADAT5:   DEFB    1,44,20             ;イベント5
457          DEFW    20*48+44+VRAMMODOKI
458          DEFW    CPATB
459          DEFW    MOVDATB
460          DEFB    0
461 ;..................................................................
462 ;
463 ;        敵キャラクタデータバッファの初期化
464 ;
465 INITWK:  LD      HL,CHRBUF           ;HL=バッファ先頭
466          LD      DE,BUFSIZ           ;DE=バッファサイズ/1機
467          LD      B,CHRMAX            ;B=敵最大数
468 INIWK0:  LD      (HL),DEAD           ;CHRBUF.TYPE=DEAD
469          ADD     HL,DE               ;ポインタを進めて
470          DJNZ    INIWK0              ;敵の最大数分繰り返す
471          RET
472 ;
473 ;        敵の移動処理
474 ;
475 MOVETEKI:
476          LD      IX,CHRBUF           ;IX=敵バッファ先頭
477          LD      B,CHRMAX            ;B=敵の最大数
478 MOVELP:  PUSH    BC                  ;ループカウンタを待避
479          LD      A,(IX+TYPE)         ;A=CHRBUF.TYPE
480          INC     A                   ;死んでいるか?
481          CALL    NZ,TEKISUB          ;生きていれば移動処理へ
482          LD      DE,BUFSIZ           ;ポインタを進め
483          ADD     IX,DE
484          POP     BC
485          DJNZ    MOVELP
486          RET
487 ;
488 TEKISUB:
489          DEC     A                   ;Aを増やしていた分を補正
490          ADD     A,A                 ;以下は典型的な
```

▶おおっ！ ひさびさに清水和人の登場だぁっ！ やっぱ，この人が書かないといまいちありがたみがありませんね。R-TYPEももちろん清水さんが書くんでしょう。ところでROGUEスゴロクで思い出したけど，私は歯に衣を着せない記事を書いていた，吉田幸一氏が好きだったんだけどなあ。
高野 勤 (19) 福島県

```
491          LD       E,A              ;  ジャンプテーブルの
492          LD       D,0              ;  利用方法
493          LD       HL,JMPTBL        ;
494          ADD      HL,DE            ;
495          LD       E,(HL)           ;
496          INC      HL               ;
497          LD       D,(HL)           ;
498          EX       DE,HL            ;
499          JP       (HL)             ;
500 ;
501 JMPTBL:  DEFW     TEKIA,TEKIB,DMY,DMY
502          DEFW     DMY,DMY,DMY,DMY
503          DEFW     DMY,DMY,DMY,DMY
504          DEFW     DMY,DMY,DMY,DMY
505 ;
506 ;        敵タイプA(直進)移動処理
507 ;
508 TEKIA:   XOR      A                ;A=向きコード0(左へ)
509          JR       MOVE&PUT         ;移動&表示
510 ;
511 ;        敵タイプB(蛇行)移動処理
512 ;
513 TEKIB:   LD       L,(IX+WORK1)     ;HL=移動データへのポインタ
514          LD       H,(IX+WORK2)     ;
515 TEKIB0:  LD       A,(HL)           ;A=向きコード
516          INC      HL               ;移動データポインタ++
517          CP       -1               ;移動データの終端か?
518          JR       NZ,TEKIB1        ;そうでなければ分岐
519          LD       HL,MOVDATB       ;HL=移動データ先頭
520          JR       TEKIB0           ;もっぺんやり直し
521 TEKIB1:  LD       (IX+WORK1),L     ;移動データポインタを更新
522          LD       (IX+WORK2),H     ;
523          CALL     MOVE&PUT         ;移動&表示
524          RET      C                ;CY=1なら死んでいる
525          LD       A,(IX+X)         ;X座標が
526          CP       20               ;  20でないのなら
527          RET      NZ               ;  リターン
528          JP       EFIRE            ;X=20なら弾を撃つ
529 ;
530 MOVDATB:                           ;敵タイプB移動データ
531          DEFB     7,0,7,0
532          DEFB     0,1,0,1
533          DEFB     1,1,0,1
534          DEFB     0,0,7,0
535          DEFB     7,7,-1           ;データの終わりは-1で表す
536 ;
537 ;        敵の移動・表示処理
538 ;
539 MOVE&PUT:
540          CALL     MOVE             ;移動して
541          RET      C                ;死んでいたら表示しない
542          CALL     PUT              ;表示する
543          OR       A
544          RET
545 ;
546 ;        向きコードによる敵移動処理
547 ;                 in        A=向きコード
548 ;                           IX=データバッファ
549 ;
550 MOVE:    ADD      A,A              ;A=向きコードx2
551          ADD      A,A              ;A=向きコードx4
552          LD       HL,DXDYDA        ;HL=変位データテーブル先頭
553          LD       E,A              ;DE=向きコードx4
554          LD       D,0              ;
555          ADD      HL,DE            ;それをHLに足す
556          LD       A,(HL)           ;A=X座標変位(=DX)
557          INC      HL
558          ADD      A,(IX+X)         ;A=X+DX
559          CP       46               ;X座標は既定範囲内か?
560          JR       NC,GOOUT         ;そうでなければ殺す
561          LD       (IX+X),A         ;X座標更新
562          LD       A,(HL)           ;A=Y座標変位(=DY)
563          INC      HL
564          ADD      A,(IX+Y)         ;A=Y+DY
565          CP       30               ;Y座標は既定範囲内か
566          JR       NC,GOOUT         ;そうでなければ殺す
567          LD       E,(HL)           ;DE=アドレス変位(=DA)
568          INC      HL               ;
569          LD       D,(HL)           ;
570          LD       L,(IX+ADRL)      ;HL=表示アドレス
571          LD       H,(IX+ADRH)      ;
572          ADD      HL,DE            ;HL=ADR+DA
573          LD       (IX+ADRL),L      ;表示アドレス更新
574          LD       (IX+ADRH),H      ;
575          OR       A                ;CY=0...生きている
576          RET
577 ;
578 ;        敵キャラを殺す
579 ;
580 GOOUT:   LD       (IX+TYPE),DEAD   ;敵タイプ=DEAD
581          SCF                       ;CY=1...死んでいる
582          RET
583 ;
584 ;        向きコードに対応する座標・アドレス変位テーブル
585 ;
586 DXDYDA:
587          DEFB     -1,0             ;0        左
588          DEFW     -1               ;
589          DEFB     -1,-1            ;1        左上
590          DEFW     -49              ;
591          DEFB     0,-1             ;2        上
592          DEFW     -48              ;
593          DEFB     1,-1             ;3        右上
594          DEFW     -47              ;
595          DEFB     1,0              ;4        右
596          DEFW     1                ;
597          DEFB     1,1              ;5        右下
598          DEFW     49               ;
599          DEFB     0,1              ;6        下
600          DEFW     48               ;
601          DEFB     -1,1             ;7        左下
602          DEFW     47               ;
603 ;..............................................................
```

```
604 ;
605 ;          敵弾データバッファ初期化
606 ;
607 INITEMSL:
608          LD       HL,EMSLBUF       ;HL=バッファ先頭
609          LD       DE,EMBFSIZ       ;DE=バッファサイズ
610          LD       B,EMSLMAX        ;B=敵弾の最大数
611 INIEMSL: LD       (HL),DEAD        ;EMSLBUF.X=DEAD
612          ADD      HL,DE            ;ポインタを進める
613          DJNZ     INIEMSL          ;敵弾の最大数分繰り返す
614          RET
615 ;
616 ;          敵弾バッファの空きエリアを探す
617 ;                 out      CY=1 ... 空き無し
618 ;                          CY=0 ... IY=空きエリアへのポインタ
619 ;
620 EMSLALC:
621          LD       IY,EMSLBUF       ;IY=バッファ先頭
622          LD       DE,EMBFSIZ       ;DE=バッファサイズ
623          LD       B,EMSLMAX        ;B=敵弾の最大数
624 EMALCL:  LD       A,(IY+EMX)       ;A=敵弾X座標
625          CP       44               ;X座標は44以上か?
626          RET      NC               ;そうなら空き領域
627          LD       A,(IY+EMY)       ;A=敵弾Y座標
628          CP       28               ;Y座標は28以上か?
629          RET      NC               ;そうなら空き領域
630          ADD      IY,DE            ;ポインタを進める
631          DJNZ     EMALCL           ;バッファの終わりまで繰り返す
632          SCF                       ;空き領域が見つからなかった
633          RET
634 ;
635 ;          敵弾の発射処理
636 ;
637 EFIRE:   CALL     EMSLALC          ;敵弾バッファの空きを探す
638          RET      C                ;CY=1なら空きが無い
639          LD       A,(IX+X)         ;X座標をセット
640          LD       (IY+EMX),A       ;
641          LD       A,(IX+Y)         ;Y座標をセット
642          LD       (IY+EMY),A       ;
643          LD       A,(IX+ADRL)      ;表示アドレスをセット
644          LD       (IY+EMADRL),A    ;
645          LD       A,(IX+ADRH)      ;
646          LD       (IY+EMADRH),A    ;
647          LD       (IY+EMP),128     ;確率の初期値
648          LD       A,(MSY)          ;自機のY座標と
649          SUB      (IX+Y)           ;  敵のY座標の差を求める
650          CALL     SGN&ABS          ;その符号と絶対値を求める
651          LD       (IY+EMDY0),B     ;B=SY
652          LD       (IY+EMDY1),B     ;
653          LD       E,A              ;E=DY
654          LD       A,(MSX)          ;自機のX座標と
655          SUB      (IX+X)           ;  敵のY座標の差を求める
656          CALL     SGN&ABS          ;その符号と絶対値を求める
657          LD       (IY+EMDX0),B     ;B=SX
658          LD       (IY+EMDX1),B     ;
659          LD       C,A              ;C=DX
660          CP       E                ;DXとDYを比べる
661          JR       Z,EFIRXY         ;等しければ45度方向
662          JR       C,EFIRY          ;DYが大きければ急な傾き
663 EFIRX:   LD       H,E              ;HL=DYx256
664          LD       L,0              ;
665                                    ;C=DX
666          CALL     DIV              ;HLをCで割る
667          LD       (IY+EMDP),L      ;確率のステップ値をセット
668          LD       (IY+EMDY0),0     ;普段はY座標は変化しない
669          JR       SETDA
670 EFIRY:   LD       H,C              ;HL=DXx256
671          LD       L,0              ;
672          LD       C,E              ;C=DY
673          CALL     DIV              ;HLをCで割る
674          LD       (IY+EMDP),L      ;確率のステップ値をセット
675          LD       (IY+EMDX0),0     ;普段はX座標は変化しない
676          JR       SETDA
677 EFIRXY:  LD       (IY+EMDP),0      ;45度方向だから毎回同じ方向
678 SETDA:   LD       E,(IY+EMDX0)     ;E=DX
679          LD       D,(IY+EMDY0)     ;D=DY
680          CALL     GETDA            ;DX,DYに対応するアドレス変位を得て
681          LD       (IY+EMDA0L),L    ;セット
682          LD       (IY+EMDA0H),H    ;
683          LD       E,(IY+EMDX1)     ;E=DX'
684          LD       D,(IY+EMDY1)     ;D=DY'
685          CALL     GETDA            ;DX',DY'に対応するアドレス変位を得
686          LD       (IY+EMDA1L),L    ;セット
687          LD       (IY+EMDA1H),H    ;
688 ;        OR       A
689          RET
690 ;
691 ;        DX,DYに対応するアドレス変位を求める
692 ;                 in       E=DX,D=DY
693 ;                 out      HL=DA
694 ;
695 GETDA:   LD       HL,DXDYDA        ;HL=テーブル先頭
696 GETDA1:  LD       A,(HL)           ;A=DXn
697          INC      HL
698          CP       E                ;DXと比べて
699          JR       NZ,GETDA2        ;一致しなければスキップ
700          LD       A,(HL)           ;A=DYn
701          INC      HL
702          CP       D                ;DYと比べて
703          JR       NZ,GETDA3        ;一致しなければスキップ
704          LD       A,(HL)           ;HLにアドレス変位を取り出す
705          INC      HL               ;
706          LD       H,(HL)           ;
707          LD       L,A              ;
708          RET
709 GETDA2:  INC      HL               ;ポインタを進める
710 GETDA3:  INC      HL
711          INC      HL
712          JR       GETDA1           ;一致するまで繰り返す
713 ;
714 ;        直前の8ビット減算結果の
715 ;        符号と絶対値を得る
```

▶僕は高校のころ，音楽部合唱班に所属していたのですが，音楽の基本的な部分はなにも知らないままに，てきとーにやっていました。でも，12月号の特集のおかげで，音楽というものが少しはわかったような気がします。　小林 到 (18) 長野県

```
716 ;              in       A,F
717 ;              out      A=ABS
718 ;                       B=SGN
719 ;
720 SGN&ABS:
721          LD       B,0             ;B=0
722          RET      Z               ;減算結果が0であれば戻る
723          JR       C,MINS          ;減算結果が負なら分岐
724          INC      B               ;B=1
725          RET                      ;SGN=1
726 MINS:    NEG      ·               ;A=-A
727          DEC      B               ;B=-1
728          RET                      ;SGN=-1
729 ;
730 ;        HL/C=HL...A
731 ;
732 DIV:     XOR      A               ;Aには余りが入る
733          LD       B,16            ;16ビットだから
734 DIV0:    ADD      HL,HL           ;被除数の最上位ビットを
735          RLA                      ;  Aレジスタに取り込む
736          INC      L               ;仮に1を立てる
737          SUB      C               ;除数を引いてみて
738          JR       NC,DIV1         ;引けたらそれでよし
739                                   ;引けなかったら
740          DEC      L               ;1は立たなかった
741          ADD      A,C             ;引きすぎたから直す
742 DIV1:    DJNZ     DIV0            ;16ビット分繰り返す
743          RET                      ;HL/C=HL...A
744 ;
745 ;        敵弾移動・表示処理
746 ;
747 MOVEEMSL:
748          LD       IX,EMSLBUF      ;IX=敵弾データバッファ
749          LD       B,EMSLMAX       ;B=敵弾の最大数
750          LD       HL,EMSLCNT      ;カウンタを進め
751          RRC      (HL)            ;
752          JR       NC,MOVEM3       ;NCであれば移動しない
753 MOVEML:  PUSH     BC              ;ループカウンタを待避
754          CALL     MOVEMC          ;死んでいるかどうかチェックして
755          JR       NC,MOVEM2       ;NCであれば死んでいる
756          LD       A,(IX+EMP)      ;確率カウンタに
757          ADD      A,(IX+EMDP)     ;  ステップ値を足して
758          LD       (IX+EMP),A      ;  更新
759          JR       C,MOVEM0        ;256を越えたなら分岐
760          LD       E,(IX+EMDX0)    ;E=DX
761          LD       D,(IX+EMDY0)    ;D=DY
762          LD       C,(IX+EMDA0L)   ;BC=DA
763          LD       B,(IX+EMDA0H)   ;
764          JR       MOVEM1
765 MOVEM0:  LD       E,(IX+EMDX1)    ;E=DX'
766          LD       D,(IX+EMDY1)    ;D=DY'
767          LD       C,(IX+EMDA1L)   ;BC=DA'
768          LD       B,(IX+EMDA1H)   ;
769 MOVEM1:  LD       A,(IX+EMX)      ;X座標を取り出し
770          ADD      A,E             ;変位を加えて
771          LD       (IX+EMX),A      ;更新
772          LD       A,(IX+EMY)      ;Y座標を取り出し
773          ADD      A,D             ;変位を加えて
774          LD       (IX+EMY),A      ;更新
775          LD       L,(IX+EMADRL)   ;表示アドレスを取り出し
776          LD       H,(IX+EMADRH)   ;
777          ADD      HL,BC           ;変位を加えて
778          LD       (IX+EMADRL),L   ;更新
779          LD       (IX+EMADRH),H   ;
780          LD       (HL),EMSLCHR    ;その位置に敵弾を表示
781 ;        ................MZ-700では以下の行を復活させる
782 ;        LD       DE,48*32        ;敵弾の
783 ;        ADD      HL,DE           ;  アトリビュートを
784 ;        LD       (HL),EMSLATR    ;  書き込む
785 ;        ................ここまでがMZ-700専用追加部分
786 MOVEM2:  LD       DE,EMBFSIZ      ;ポインタを
787          ADD      IX,DE           ;進めて
788          POP      BC              ;ループカウンタを取り出し
789          DJNZ     MOVEML          ;バッファの終わりまで繰り返す
790          RET
791 ;
792 MOVEM3:  LD       DE,EMBFSIZ      ;バッファサイズ/1発
793 MOVEM4:  CALL     MOVEMC          ;死んでいるかどうかチェックして
794          JR       NC,MOVEM5       ;NCであれば死んでいる
795          LD       L,(IX+EMADRL)   ;表示アドレスを取り出し
796          LD       H,(IX+EMADRH)   ;
797          LD       (HL),EMSLCHR    ;その位置に敵弾を表示
798 ;        ................MZ-700では以下の行を復活させる
799 ;        PUSH     DE
800 ;        LD       DE,48*32        ;敵弾の
801 ;        ADD      HL,DE           ;  アトリビュートを
802 ;        LD       (HL),EMSLATR    ;  書き込む
803 ;        POP      DE
804 ;        ................ここまでがMZ-700専用追加部分
805 MOVEM5:  ADD      IX,DE           ;ポインタを進めて
806          DJNZ     MOVEM4          ;バッファの終わりまで繰り返す
807          RET
808 ;
809 MOVEMC:  LD       A,(IX+EMX)      ;X座標を取り出し
810          CP       44              ;最大値以上なら
811          RET      NC              ;死んでいる
812          LD       A,(IX+EMY)      ;Y座標を取り出し
813          CP       28              ;最大値以上なら
814                                   ;死んでいる
815          RET
816 ;.....................................................................
817 ;
818 ;        2x2キャラの仮想VRAMへの書き込み
819 ;
820 PUT:     LD       E,(IX+ADRL)     ;DE=キャラ左上の表示アドレス
821          LD       D,(IX+ADRH)     ;
822          LD       L,(IX+PATL)     ;HL=キャラパターン
823          LD       H,(IX+PATH)     ;
824 ;        ................MZ-700では以下の行を復活する
825 ;        PUSH     DE              ;表示アドレスを保存
826 ;        CALL     PUT0            ;仮想テキストVRAMへ書き込み
827 ;        POP      DE              ;表示アドレスを復帰
828 ;        EX       DE,HL           ;DE=DE+48*32
829 ;        LD       BC,48*32        ;
830 ;        ADD      HL,BC           ;
831 ;        EX       DE,HL           ;
832 ;        ................ここまでがMZ-700専用追加部分
833 PUT0:    LDI                      ;左上
834          LDI                      ;右上
835          EX       DE,HL           ;DE=キャラ左下の表示アドレス
836          LD       BC,46           ;
837          ADD      HL,BC           ;
838          EX       DE,HL           ;
839          LDI                      ;左下
840          LDI                      ;右下
841          RET
842 ;.....................................................................
843 ;
844 ;        一時停止処理
845 ;
846 PAUSE:   CALL     INKEY           ;キー入力
847          RL       A               ;トリガー2はON?
848          RET      M               ;そうでなければ戻る
849 PAUSE0:  CALL     INKEY           ;キー入力
850          AND      040H            ;トリガー2が
851          JR       Z,PAUSE0        ;  離されるのを待つ
852 PAUSE1:  CALL     INKEY           ;キー入力
853          CPL                      ;ビット反転
854          AND      060H            ;トリガー1か2が
855          JR       Z,PAUSE1        ;  押されるのを待つ
856 PAUSE2:  CALL     INKEY           ;キー入力
857          RL       A               ;トリガー2が
858          RET      M               ;  離されるのを
859          JR       PAUSE2          ;  待つ
860 ;.....................................................................
861 ;
862 ;        ワークエリア
863 ;
864 MSBUF:                            ;自機データバッファ
865 MSFLAG:  DEFS     1               ;  フラグ
866 MSX:     DEFS     1               ;  X座標
867 MSY:     DEFS     1               ;  Y座標
868 MSADR:   DEFS     2               ;  表示アドレス
869 MSPAT:   DEFS     2               ;  キャラクタパターンへのポインタ
870 MSWK1:   DEFS     1               ;  直前のトリガーの状態
871 MSWK2:   DEFS     1               ;  トリガーリピートカウンタ
872 MSWK3:   DEFS     1               ;  未使用
873 ;
874 EMSLCNT:
875          DEFS     1
876 MMSLBUF:                          ;自弾データバッファ
877          DEFS     MBUFSIZ*MMSLMAX
878 CHRBUF:                           ;敵キャラデータバッファ
879          DEFS     BUFSIZ*CHRMAX
880 EMSLBUF:                          ;敵弾データバッファ
881          DEFS     EMBFSIZ*EMSLMAX
882 VRAMMODOKI:                       ;48x32仮想テキストVRAM
883          DEFS     48*32
884 ;        ................MZ-700では以下の行を復活させる
885 ;AVRAMMODOKI:                     ;48x32仮想アトリビュートVRAM
886 ;        DEFS     48*32
887 ;        ................ここまでがMZ-700専用追加部分
888 PRGEND:
889 ;---------------------------------------------------------------------
890 ;---------------------------------------------------------------------
891 ;        マップ/イベントデータ  サンプル
892 ;
893
894 ;
895 ;        イベントデータテーブル  40バイトで1画面分
896 ;
897 IVNTDAT:
898          DEFB     1,1,0,0,2,2,0,0,3,3,3,3,0,0,4,4
899          DEFB     4,4,0,0,5,5,5,5,0,0,0,0,1,1,0,0
900          DEFB     2,2,0,0,3,3,3,3
901
902          DEFB     0,0,4,4,4,4,0,0,5,5,5,5,0,0,0,0
903          DEFB     1,1,0,0,2,2,0,0,3,3,3,3,0,0,4,4
904          DEFB     4,4,0,0,5,5,5,5
905
906          DEFB     80H,0,0,0,1,1,0,0,2,2,0,0,3,3,3,3
907          DEFB     0,0,4,4,4,4,0,0,5,5,5,5,0,0,0,0
908          DEFB     1,1,0,0,2,2,0,0
909
910          DEFB     3,3,3,3,0,0,4,4,4,4,0,0,5,5,5,5
911          DEFB     0,0,0,0,1,1,0,0,2,2,0,0,3,3,3,3
912          DEFB     0,0,4,4,4,4,0,0
913
914          DEFB     -1                         ;終端は-1で表す
915 ;
916 ;        マップデータテーブル    60バイトで1画面分
917 ;        (繰り返し使う時は最初と最後の
918 ;                    ブロックを同じにしておく)
919 ;
920 ;
921 MAPDAT:  DEFB     0,0,0,1,0,4,2,2,2,2,2,4
922          DEFB     0,0,0,0,0,4,2,2,3,2,2,4
923          DEFB     0,0,0,0,0,4,2,2,2,2,2,4
924          DEFB     5,1,0,0,1,4,5,2,2,2,2,4
925          DEFB     5,0,0,0,0,4,5,2,2,3,2,4
926
927          DEFB     5,0,0,0,5,5,5,5,2,3,2,5
928          DEFB     5,0,0,0,5,5,5,5,2,2,2,5
929          DÈFB     0,0,0,0,0,5,1,2,2,2,2,5
930          DEFB     0,0,0,0,0,5,2,3,3,2,2,5
931          DEFB     0,0,0,0,0,5,2,2,2,2,2,5
932
933          DEFB     0,0,0,1,0,4,2,2,2,2,2,4
934          DEFB     0,0,0,0,0,4,2,2,3,2,2,4
935          DEFB     0,0,0,0,0,4,2,2,2,2,2,4
936          DEFB     5,1,0,0,1,4,5,2,2,2,2,4
937          DEFB     5,0,0,0,0,4,5,2,2,3,2,4
938
939          DEFB     5,0,0,0,5,5,5,5,2,3,2,5
940 MAPEND:
941 ;
942 ;---------------------------------------------------------------------
```

THE SENTINEL

第75部 パズルゲームLAST ONE
第76部 ブロックゲームFLICK

●パズルゲーム２本立て

皆さんあけましておめでとうございます。今年もS-OSをよろしくお願いいたします。ということで，新年はまずパズルゲームでS-OS初めといきましょう。今回はボード版ソリティア風仕立てのLAST ONEとユニークなブロックタイプパズルゲームFLICKの２本立てです。どちらも比較的短いプログラムですから，気軽に入力してみてください。

こういったパズルゲームは一度，解法がわかってしまったら，楽しみはなくなってしまうのですが，LAST ONEはコンストラクション機能つき，またFLICKも簡単に自作の面を構成していくことができますので全面クリアした人は(LAST ONEにはひとつだけ絶対に解けない面も含まれています）オリジナル面を開発して友だちとデータ交換するなどというぐあいに楽しみ方を広げていくとよいでしょう。

●ついに新アセンブラ登場か

お待たせしました。来月ではいよいよZEDAに代わる新エディタアセンブラを発表する予定です。

これまでに，S-OS用としていろいろな言語が現れましたが，まだまだアセンブラがS-OSの標準言語のようです。やはり，限られたメモリ空間の中でマシンの性能いっぱいのことを行おうとすると，どうしてもマシン語を避けて通るわけにはいきません。

初心者にはものすごく難しいと思われているマシン語もちゃんとステップを踏んでいけば，案外簡単なものだと思われる方もいるかもしれません。これまでアセンブラが入手できなくてマシン語でのプログラミングに親しんでいなかった人も，これを機会に究極の言語，アセンブラを修得してみてください。

全機種共通システム掲載記事

■85年６月号
序論　共通化の試み
第１部　S-OS"MACE"
第２部　Lisp-85インタプリタ
第３部　チェックサムプログラム
■85年７月号
第４部　マシン語プログラム開発入門
第５部　エディタアセンブラZEDA
第６部　デバッグツールZAID
■85年８月号
第７部　ゲーム開発パッケージBEMS
第８部　ソースジェネレータZING
■85年９月号
インタラプト　S-OS番外地
第９部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年１月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年２月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年３月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年４月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年５月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年６月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　"SWORD"2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年７月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS"SWORD"
■86年８月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS"SWORD"
■86年９月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>
■87年１月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年２月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年３月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　"SWORD"再掲載とMAGICの標準化
■87年４月号
第40部　INVADER GAME
第41部　TANGERINE
■87年５月号
第42部　S-OS"SWORD"変身セット
第43部　MZ-700用"SWORD"をQD対応に
■87年６月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年７月号
第46部　STORY MASTER
■87年８月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS"SWORD"
■87年９月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版"SWORD"アフターケア ラインプリントルーチン
特別付録　PASOPIA7版S-OS"SWORD"
■88年１月号
第58部　Fuzzy BASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年２月号
第59部　シューティングゲームELFES
■88年３月号
第60部　構造型コンパイラ言語SLANG
■88年４月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年５月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年６月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年７月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年８月号
第68部　マルチウィンドウエディタWINER
■88年９月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY

＊以上のアプリケーションは，基本システムであるS-OS"MACE"またはS-OS"SWORD"がないと動作しませんのでご注意ください。

全種種共通S-OS“SWORD”要

パズルゲームLAST ONE

久しぶりにS-OS用のパズルゲームの登場です。ずらりと並んだピンを飛び越えて，ボード上にひとつだけピンを残すようにピンを取り去っていきます。あなたはいくつ面をクリアできるでしょうか。コンストラクション機能付きで楽しさも広がります。

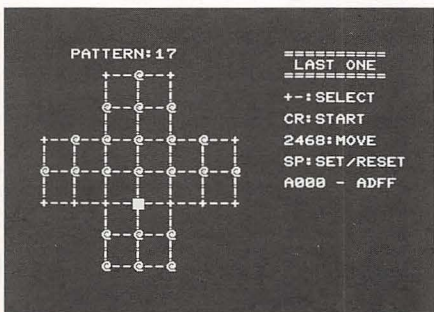
中島　聡 *Nakajima Satoshi*

はじめに

昔LSIゲームにあったボード状のパズルゲームを参考にS-OS版を作ってみました。パズルに詳しい方ならソリティアという名前のほうがわかりやすいかもしれません。古典的なゲームでルールもシンプルですが，なかなか味わい深いものがあります。

最近はシューティングゲームが多くて，S-OSのパズルゲームは久しぶりなのですが，プログラムサイズも3Kバイトほどですので手軽に打ち込んでパズルを楽しんでみてください。なお，入力の際は各機種のマシン語モニタまたはMACINTO-Cなどのマシン語入力ツールを使ってリスト1を打ち込んでください。

遊び方&ルール解説

プログラムはリロケータブルに組まれていますので適当な番地から読み込んで使ってください。最初はデモモードになっていますがSキーを押すとゲーム開始です。画面には十文字型の盤が表示され，その上にさまざまな配置で駒（@）が並んでいます。まず，どの面から始めるかを決めます。＋とーのキーで順次駒の配置が変わっていきますので，自分のやりたい面を選択してリターンキーで決定してください。

点滅しているカーソルはテンキーの2，4, 6, 8またはカーソルキーによって移動していきます。ここで基本的なルールを説明しておきましょう。

画面上の駒は隣に並んでいる駒を飛び越して移動していきます。カーソルを移動してスペースキーで駒を選択し，移動方向にキーを押すことで駒を動かしてください。このとき飛び越された駒は盤上から取り除かれます。駒の移動は必ずほかの駒を飛び越すかたちでなくてはなりません（移動だけはできない)。たとえば，

```
    @@    @@
```

のような配置で駒が並んでいるときは次の1手として4通りの手が考えられます。すなわち，

```
1)      @   @@
2)  @         @@
3)     @@   @
4)     @@       @
```

の4つです。

しかし，その後の展開から2)，4)は適当ではありません。このまま1)を進めると，

```
      @   @@
        @@
    @
```

のようになります。こうしていくと最後には盤上に駒がひとつだけ残りますね。これが画面の中央に残っていれば1面クリアです。画面の中央以外の場所に残ってしまったら“REGRETFUL”と表示されますので，もう一度挑戦してみてください。実際にはこれが2次元で配置されていますのでもう少し複雑になります。

これを読んだだけではよくわからないかもしれませんが，ゲームを起動するとオートデモが始まりますのでそれをじっくり見てください。

コンストラクション

面は0から29までの30面用意してあります。途中で行き詰まったときはリターンキーを押すと最初からやりなおすことができます。30面以降を選択したときは“YOU CAN EDIT.”と表示されエディットモードに入ります。カーソルを移動しスペースキーで駒のセット/リセットを行ってください。ひとつの面を作るとプログラムサイズが64バイトずつ増えていきます。それは画面の右側に常に表示されていますので，そのとおりにセーブしなおせば創作パターンの保存ができます。また，エディット中にCキーを押すとその面以降のパターンは消去されますので注意してください。

Profile

◇中島さんは福岡県にお住まいの21歳，大学3年生です。マイコン歴は約5年のX1ユーザー。さて，問題です。このゲームで解けない面は何面なのでしょうか。

リスト1　LAST ONEダンプリスト

```
A000 CD F7 1F 7D FE 20 C2 C4 : 04
A008 1F 21 39 20 11 00 00 01 : AB
A010 09 00 CD 97 1F CD 80 1F : F8
A018 11 E8 FF 19 E5 DD E1 11 : C5
A020 89 05 DD 19 DD 22 3A 20 : DD
A028 DD 75 09 DD 74 0A 11 10 : D7
A030 00 DD 19 DD 22 3D 20 DD : 2F
A038 75 07 DD 74 08 11 0E 00 : F4
A040 DD 19 DD 22 40 20 DD 75 : A7
A048 09 DD 74 0A 3E C3 32 39 : D0
A050 20 32 3C 20 32 3F 20 CD : 0C
A058 D6 1F CD 3F 20 B8 05 D5 : B3
A060 DD E1 DD 36 00 00 CD 39 : D7
A068 20 5C 02 CD 39 20 6C 03 : 13
A070 21 13 11 CD 1E 20 CD E2 : FF
A078 1F 44 45 4D 4F 4E 53 54 : 39
-----------------------------------
SUM: FA 39 8F 3C 04 AC 29 C4 DC04
```

```
A080 52 41 54 49 4F 4E 00 21 : EE
A088 13 13 CD 1E 20 CD E2 1F : FF
A090 50 52 45 53 53 20 5B 53 : 5B
A098 5D 20 4B 45 59 00 CD 3F : 72
A0A0 20 B5 00 2A 22 20 E5 ED : 13
A0A8 53 22 20 CD 3F 20 BB 05 : 81
A0B0 D5 FD E1 18 28 CD 39 20 : 19
A0B8 49 05 FE 53 28 11 FD 7E : 53
A0C0 00 FD 23 B7 C0 CD 3F 20 : C3
A0C8 BB 05 D5 FD E1 18 EF E1 : 5B
A0D0 E1 E1 22 22 20 CD 39 20 : 4C
A0D8 5C 02 DD 34 00 CD 39 20 : 95
A0E0 6C 03 30 35 CD 39 20 FC : F6
A0E8 03 CD 39 20 74 05 FE 2B : CB
A0F0 28 0A FE 2D 28 14 FE 0D : A4
A0F8 28 69 18 E8 DD 7E 01 DD : CA
-----------------------------------
SUM: 5A C7 26 D5 D3 A8 9D B4 11DE
```

```
A100 34 00 DD BE 00 30 D6 AF : 84
A108 18 0A DD 7E 00 D6 01 30 : 84
A110 03 DD 7E 01 DD 77 00 18 : CB
A118 C4 CD 39 20 E9 03 CD 39 : DC
A120 20 FC 03 CD 39 20 74 05 : BE
A128 FE 2B 28 D0 FE 2D 28 DA : 4E
A130 FE 0D 28 2A FE 20 28 10 : B3
A138 FE 43 20 05 CD 3C 20 0B : 9A
A140 05 CD 39 20 13 04 18 D6 : 30
A148 CD 1B 20 EE 6B CD F4 1F : 41
A150 CD C4 1F CD 39 20 07 04 : E1
A158 7E EE 6B 77 18 C0 CD 39 : 2C
A160 20 EE 04 CD 39 20 91 03 : CC
A168 CD 39 20 DD 04 CD 39 20 : 2D
A170 FC 03 CD 39 20 74 05 FE : 9C
A178 2B 28 81 FE 2D 28 8B FE : B0
-----------------------------------
SUM: 5E 17 39 5C 21 63 C2 7B 776F
```

```
A180 0D 28 E0 FE 20 28 07 CD : 2F
A188 39 20 13 04 18 DF CD 1B : 4F
A190 20 FE 40 28 08 CD C4 1F : 3E
A198 CD C4 1F 18 D0 3E 7B CD : 1E
A1A0 F4 1F CD C4 1F CD 39 20 : E9
A1A8 FC 03 CD 39 20 74 05 FE : 9C
A1B0 20 20 07 3E 40 CD F4 1F : A5
A1B8 18 DE C5 CD 39 20 62 04 : 47
A1C0 30 06 C1 CD C4 1F 18 DA : 99
A1C8 E1 C5 44 4D CD 39 20 FC : 59
A1D0 03 3E 2B CD F4 1F EB CD : 04
A1D8 1E 20 3E 2B CD F4 1F C1 : 48
A1E0 CD 39 20 FC 03 3E 40 CD : 70
A1E8 F4 1F CD C4 1F C5 16 00 : 9E
A1F0 0E 07 06 07 CD 39 20 FC : 44
A1F8 03 CD 1B 20 FE 40 20 01 : 6A
---------------------------------
SUM: 5F 7F 34 43 07 27 7F 43 EDAA

A200 14 10 F1 0D 20 EC C1 15 : 04
A208 28 05 CD 3C 20 6D 01 21 : E5
A210 0C 0C CD 1B 20 47 21 09 : 91
A218 17 CD 1E 20 78 FE 40 20 : F8
A220 18 CD E2 1F 43 4F 4E 47 : 0D
A228 52 41 54 55 4C 41 54 49 : 66
A230 4F 4E 53 21 21 20 00 18 : 6A
A238 10 CD E2 1F 52 45 47 52 : 0E
A240 45 54 46 55 4C 21 21 20 : E2
A248 00 CD 39 20 DD 04 CD 39 : 0D
A250 20 74 05 FE 0D 20 F7 CD : 88
A258 3C 20 DD 00 AF CD 30 20 : 05
A260 21 06 01 CD 1E 20 CD E2 : E2
A268 1F 50 41 54 54 45 52 4E : 3D
A270 3A 00 21 1A 01 CD 1E 20 : 81
A278 CD E2 1F 3D 3D 3D 3D 3D : FF
---------------------------------
SUM: 10 04 F7 23 6F 14 9B 2C 41A8

A280 3D 3D 3D 3D 3D 00 21 1B : 6D
A288 02 CD 1E 20 CD E2 1F 4C : 27
A290 41 53 54 20 4F 4E 45 00 : EA
A298 21 1A 03 CD 1E 20 CD E2 : F8
A2A0 1F 3D 3D 3D 3D 3D 3D 3D : CA
A2A8 3D 3D 3D 00 21 1A 05 CD : C4
A2B0 1E 20 CD E2 1F 2B 2D 3A : 9E
A2B8 53 45 4C 45 43 54 00 21 : E1
A2C0 1A 07 CD 1E 20 CD E2 1F : FA
A2C8 43 52 3A 53 54 41 52 54 : 5D
A2D0 00 21 1A 09 CD 1E 20 CD : 1C
A2D8 E2 1F 32 34 36 38 3A 4D : 5C
A2E0 4F 56 45 00 21 1A 0B CD : FD
A2E8 1E 20 CD E2 1F 53 50 3A : E9
A2F0 53 45 54 2F 52 45 53 45 : 4A
A2F8 54 00 21 09 03 CD 39 20 : A7
---------------------------------
SUM: C1 AA 1F 76 43 09 36 A7 A906

A300 23 03 21 03 09 CD 39 20 : 79
A308 23 03 21 09 09 CD 39 20 : 7F
A310 23 03 21 0F 09 CD 39 20 : 85
A318 23 03 21 09 0F CD 39 20 : 85
A320 23 03 C9 CD 3F 20 3B 03 : 59
A328 0E 07 CD 1E 20 06 07 1A : 47
A330 13 CD F4 1F 10 F9 24 0D : 2D
A338 20 F0 C9 2B 2D 2D 2B 2D : B6
A340 2D 2B 21 20 20 21 20 20 : 1A
A348 21 21 20 20 21 20 20 21 : 04
A350 2B 2D 2D 2B 2D 2D 2B 21 : 56
A358 20 20 21 20 20 21 21 20 : 03
A360 20 21 20 20 21 2B 2D 2D : 27
A368 2B 2D 2D 2B DD 7E 00 FE : 09
A370 1E F5 38 1E 21 09 17 CD : 77
A378 1E 20 CD E2 1F 59 4F 55 : 09
---------------------------------
SUM: 10 CF B8 2F 92 1A 94 A6 A536

A380 20 43 41 4E 20 45 44 49 : E4
A388 54 20 20 20 20 20 00 18 : 0C
A390 0E F5 21 09 17 CD 1E 20 : 4F
A398 06 11 CD F1 1F 10 FB CD : CC
A3A0 39 20 25 05 21 0E 01 CD : 80
A3A8 1E 20 CD 39 20 E9 03 DD : 2D
A3B0 4E 00 06 08 AF CB 21 17 : 0E
A3B8 FE 0A 38 03 0C D6 0A 10 : 3F
A3C0 F4 CB 21 CB 21 CB 21 CB : 83
A3C8 21 B1 CD C1 1F 0E 07 06 : 9A
A3D0 07 CD 39 20 FC 03 CD 39 : 32
A3D8 20 07 04 7E CD F4 1F 10 : 99
A3E0 F0 0D 20 EB 01 01 03 F1 : FE
A3E8 C9 DD 6E 00 26 00 29 29 : 8C
A3F0 29 29 29 29 CD 3F 20 49 : 19
A3F8 06 19 EB C9 78 87 80 67 : B9
---------------------------------
SUM: 4F 2F 4C B8 E7 71 6C 03 5615

A400 79 87 81 6F C3 1E 20 78 : 69
A408 87 87 87 81 D6 09 26 00 : 1B
A410 6F 19 C9 FE 32 28 22 FE : C9
A418 34 28 27 FE 36 28 2A FE : 07
A420 38 28 0F FE 1F 28 12 FE : C4
A428 1D 28 17 FE 1C 28 1A FE : B6
A430 1E C0 05 CD 39 20 52 04 : 5F
A438 D0 04 CD 39 20 52 04 D0 : 20
A440 05 C9 0D CD 39 20 52 04 : 57
A448 D0 0C CD 39 20 52 04 D0 : 28
A450 0D C9 CD 39 20 FC 03 CD : C8
A458 1B 20 FE 2B C8 FE 40 C8 : 32
A460 37 C9 FE 32 28 1E FE 34 : A8
A468 28 24 FE 36 28 2A FE 38 : 08
A470 28 30 FE 1F 28 0E FE 1D : C6
A478 28 14 FE 1C 28 1A FE 1E : B4
---------------------------------
SUM: 92 52 8D FB 76 15 A5 54 AB74

A480 28 20 37 C9 04 CD 39 20 : 72
A488 AC 04 D8 04 18 2D 0D CD : AB
A490 39 20 AC 04 D8 0D 18 23 : 29
A498 0C CD 39 20 AC 04 D8 0C : C6
A4A0 18 19 05 CD 39 20 AC 04 : 0C
A4A8 D8 05 18 0F CD 39 20 FC : 26
A4B0 03 54 5D CD 1B 20 FE 40 : FA
A4B8 C8 37 C9 CD 39 20 FC 03 : ED
A4C0 CD 1B 20 FE 2B C8 37 C9 : F9
A4C8 CD 39 20 CD 04 CD 39 20 : 1D
A4D0 D2 04 CD 39 20 D7 04 3E : 15
A4D8 00 3D 20 FD C9 C5 06 04 : F2
A4E0 CD C4 1F CD 39 20 C8 04 : A2
A4E8 10 F6 C1 C3 C4 1F DD 7E : C8
A4F0 01 DD BE 00 C0 FE 63 C8 : 85
A4F8 DD 34 01 CD 39 20 E9 03 : 24
---------------------------------
SUM: FB 1A 03 C5 08 32 67 D7 C1CA

A500 01 40 00 EB 09 EB ED 42 : 4F
A508 ED B0 C9 CD C4 1F DD 7E : 71
A510 00 3D DD 77 00 DD 77 01 : E6
A518 CD 39 20 EE 04 DD 34 00 : 29
A520 CD 3C 20 DD 00 21 1A 0D : 4E
A528 CD 1E 20 CD 3F 20 00 00 : 37
A530 EB CD BE 1F CD E2 1F 20 : 83
A538 2D 20 00 DD 6E 01 2C CD : 92
A540 39 20 EC 03 1B EB C3 BE : CF
A548 1F C5 D5 E5 CD 18 20 CD : 70
A550 1B 20 57 01 02 00 3E 7B : 4E
A558 CD F4 1F CD 1E 20 CD 39 : F1
A560 20 D7 04 CD D0 1F FE 53 : 08
A568 28 06 10 F2 7A 0D 20 E8 : BF
A570 E1 D1 C1 C9 CD 21 20 FE : 48
A578 1B C0 21 39 20 11 00 00 : 66
---------------------------------
SUM: F1 14 F1 3A 8A 69 06 33 509B

A580 01 09 00 CD 91 1F C3 FD : 47
A588 1F 08 D9 E1 5E 23 56 23 : DB
A590 E5 21 00 00 19 E5 D9 08 : E5
A598 C9 08 D9 E1 5E 23 56 21 : 83
A5A0 00 00 19 E5 D9 08 C9 08 : B0
A5A8 D9 E1 5E 23 56 23 E5 21 : BA
A5B0 00 00 19 E5 D9 08 D1 C9 : 79
A5B8 00 1E 00 2A 0D 36 36 36 : F7
A5C0 32 20 36 34 34 32 32 20 : 74
A5C8 38 34 20 36 36 20 34 2A : 76
A5D0 2A 2A 0D 00 00 00 00 00 : 61
A5D8 00 00 00 00 00 00 00 00 : 00
A5E0 00 00 00 00 00 00 00 00 : 00
A5E8 00 00 00 00 00 00 00 00 : 00
A5F0 00 00 00 00 00 00 00 00 : 00
A5F8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 3B B7 A5 10 E5 05 63 BB F7DA

A600 00 00 00 00 00 00 00 00 : 00
A608 00 20 20 2B 2B 2B 20 20 : 01
A610 20 20 20 2B 2B 2B 20 20 : 21
A618 20 2B 2B 2B 2B 2B 2B 2B : 4D
A620 20 2B 2B 2B 2B 2B 2B 2B : 4D
A628 20 2B 2B 2B 2B 2B 2B 2B : 4D
A630 20 20 20 2B 2B 2B 20 20 : 21
A638 20 20 20 2B 2B 2B 20 20 : 21
A640 20 20 20 20 20 20 20 20 : 00
A648 20 20 20 2B 2B 2B 20 20 : 21
A650 20 20 20 2B 2B 2B 20 20 : 21
A658 20 2B 2B 2B 2B 2B 2B 2B : 4D
A660 20 2B 2B 40 40 40 2B 2B : 8C
A668 20 2B 2B 2B 40 2B 2B 2B : 62
A670 20 20 20 2B 40 2B 20 20 : 36
A678 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: C0 22 22 8F B9 8F 22 22 BA0D

A680 20 20 20 20 20 20 20 20 : 00
A688 20 20 20 2B 2B 2B 20 20 : 21
A690 20 20 20 2B 40 2B 20 20 : 36
A698 20 2B 2B 40 40 40 2B 2B : 8C
A6A0 20 2B 2B 2B 40 2B 2B 2B : 62
A6A8 20 2B 2B 2B 40 2B 2B 2B : 62
A6B0 20 20 20 2B 2B 2B 20 20 : 21
A6B8 20 20 20 2B 2B 2B 20 20 : 21
A6C0 20 20 20 20 20 20 20 20 : 00
A6C8 20 20 20 2B 2B 2B 20 20 : 21
A6D0 20 20 20 2B 2B 2B 20 20 : 21
A6D8 20 2B 2B 2B 40 2B 2B 2B : 62
A6E0 20 2B 2B 40 40 40 2B 2B : 8C
A6E8 20 2B 40 40 40 40 40 2B : B6
A6F0 20 20 20 2B 2B 2B 20 20 : 21
A6F8 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: 00 42 57 D9 2D D9 57 42 9573

A700 20 20 20 20 20 20 20 20 : 00
A708 20 20 20 2B 2B 2B 20 20 : 21
A710 20 20 20 2B 2B 2B 20 20 : 21
A718 20 2B 40 40 2B 40 40 2B : A1
A720 20 2B 2B 2B 40 2B 2B 2B : 62
A728 20 2B 2B 2B 2B 2B 2B 2B : 4D
A730 20 20 20 2B 2B 2B 20 20 : 21
A738 20 20 20 2B 2B 2B 20 20 : 21
A740 20 20 20 20 20 20 20 20 : 00
A748 20 20 20 2B 2B 2B 20 20 : 21
A750 20 20 20 2B 40 2B 20 20 : 36
A758 20 2B 2B 40 40 40 2B 2B : 8C
A760 20 2B 2B 2B 40 2B 2B 2B : 62
A768 20 2B 40 40 40 40 40 2B : B6
A770 20 20 20 2B 2B 2B 20 20 : 21
A778 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: 00 42 6C D9 03 D9 6C 42 1624

A780 20 20 20 20 20 20 20 20 : 00
A788 20 20 20 40 40 40 20 20 : 60
A790 20 20 20 40 40 40 20 20 : 60
A798 20 2B 2B 40 40 40 2B 2B : 8C
A7A0 20 2B 40 40 40 40 40 2B : B6
A7A8 20 2B 2B 2B 2B 2B 2B 2B : 4D
A7B0 20 20 20 2B 2B 2B 20 20 : 21
A7B8 20 20 20 2B 2B 2B 20 20 : 21
A7C0 20 20 20 20 20 20 20 20 : 00
A7C8 20 20 20 2B 2B 2B 20 20 : 21
A7D0 20 20 20 2B 40 2B 20 20 : 36
A7D8 20 2B 2B 40 40 40 2B 2B : 8C
A7E0 20 2B 40 40 40 40 40 2B : B6
A7E8 20 2B 2B 40 2B 40 2B 2B : 77
A7F0 20 20 20 2B 2B 2B 20 20 : 21
A7F8 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: 00 42 6C 2D 2D 2D 6C 42 D123

A800 20 20 20 20 20 20 20 20 : 00
A808 20 20 20 40 2B 40 20 20 : 4B
A810 20 20 20 40 40 40 20 20 : 60
A818 20 40 40 2B 40 2B 40 40 : B6
A820 20 2B 40 40 40 40 40 2B : B6
A828 20 40 40 2B 40 2B 40 40 : B6
A830 20 20 20 40 40 40 20 20 : 60
A838 20 20 20 40 2B 40 20 20 : 4B
A840 20 20 20 20 20 20 20 20 : 00
A848 20 20 20 2B 2B 2B 20 20 : 21
A850 20 20 20 2B 40 2B 20 20 : 36
A858 20 2B 2B 40 40 40 2B 2B : 8C
A860 20 2B 40 40 2B 40 40 2B : A1
A868 20 2B 2B 40 40 40 2B 2B : 8C
A870 20 20 20 2B 40 2B 20 20 : 36
A878 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: 00 6C 96 42 57 42 96 6C EFBB

A880 20 20 20 20 20 20 20 20 : 00
A888 20 20 20 2B 40 2B 20 20 : 36
A890 20 20 20 40 40 40 20 20 : 60
A898 20 2B 40 40 40 40 40 2B : B6
A8A0 20 40 40 40 40 40 40 40 : E0
A8A8 20 2B 2B 40 2B 40 2B 2B : 77
A8B0 20 20 20 2B 2B 2B 20 20 : 21
A8B8 20 20 20 2B 2B 2B 20 20 : 21
A8C0 20 20 20 20 20 20 20 20 : 00
A8C8 20 20 20 2B 2B 2B 20 20 : 21
A8D0 20 20 20 2B 2B 2B 20 20 : 21
A8D8 20 2B 2B 2B 2B 2B 2B 2B : 4D
A8E0 20 2B 2B 2B 40 2B 2B 2B : 62
A8E8 20 2B 2B 40 40 40 2B 2B : 8C
A8F0 20 20 20 40 40 40 20 20 : 60
A8F8 20 20 20 40 40 40 20 20 : 60
---------------------------------
SUM: 00 57 6C 2D 42 2D 6C 57 8744

A900 20 20 20 20 20 20 20 20 : 00
A908 20 20 20 2B 2B 2B 20 20 : 21
A910 20 20 20 2B 40 2B 20 20 : 36
A918 20 2B 2B 40 40 40 2B 2B : 8C
A920 20 2B 40 40 40 40 40 2B : B6
A928 20 40 40 40 40 40 40 40 : E0
A930 20 20 20 2B 2B 2B 20 20 : 21
A938 20 20 20 2B 2B 2B 20 20 : 21
A940 20 20 20 20 20 20 20 20 : 00
A948 20 20 20 2B 2B 2B 20 20 : 21
A950 20 20 20 2B 40 2B 20 20 : 36
A958 20 2B 2B 40 40 40 2B 2B : 8C
A960 20 2B 2B 40 40 40 2B 2B : 8C
A968 20 2B 2B 40 40 40 2B 2B : 8C
A970 20 20 20 2B 40 2B 20 20 : 36
A978 20 20 20 2B 2B 2B 20 20 : 21
---------------------------------
SUM: 00 57 6C 18 57 18 6C 57 2F17

A980 20 20 20 20 20 20 20 20 : 00
A988 20 20 20 2B 2B 2B 20 20 : 21
A990 20 20 20 2B 2B 2B 20 20 : 21
A998 20 2B 2B 2B 40 2B 2B 2B : 62
A9A0 20 2B 40 40 40 40 40 2B : B6
A9A8 20 2B 2B 40 40 40 2B 2B : 8C
A9B0 20 20 20 40 2B 40 20 20 : 4B
A9B8 20 20 20 2B 2B 2B 20 20 : 21
A9C0 20 20 20 20 20 20 20 20 : 00
A9C8 20 20 20 2B 40 2B 20 20 : 36
A9D0 20 20 20 2B 40 2B 20 20 : 36
A9D8 20 2B 40 40 40 40 40 2B : B6
A9E0 20 40 40 40 40 40 40 40 : E0
A9E8 20 2B 40 40 40 40 40 2B : B6
A9F0 20 20 20 2B 40 2B 20 20 : 36
A9F8 20 20 20 2B 40 2B 20 20 : 36
---------------------------------
SUM: 00 57 96 18 6C 18 96 57 9C2E

AA00 20 20 20 20 20 20 20 20 : 00
AA08 20 20 20 2B 2B 2B 20 20 : 21
AA10 20 20 20 2B 40 2B 20 20 : 36
AA18 20 2B 2B 40 40 40 2B 2B : 8C
AA20 20 40 40 40 40 40 40 40 : E0
AA28 20 2B 2B 40 40 40 2B 2B : 8C
AA30 20 20 20 2B 40 2B 20 20 : 36
AA38 20 20 20 2B 40 2B 20 20 : 36
AA40 20 20 20 20 20 20 20 20 : 00
AA48 20 20 20 2B 40 2B 20 20 : 36
AA50 20 20 20 40 40 40 20 20 : 60
AA58 20 40 40 2B 40 2B 40 40 : B6
AA60 20 2B 40 40 40 40 40 2B : B6
AA68 20 2B 2B 40 40 40 2B 2B : 8C
AA70 20 20 20 40 40 40 20 20 : 60
AA78 20 20 20 2B 40 2B 20 20 : 36
---------------------------------
SUM: 00 6C 81 2D AB 2D 81 6C 788C
```

```
AA80 20 20 20 20 20 20 20 20 : 00
AA88 20 20 20 2B 40 2B 20 20 : 36
AA90 20 20 20 40 40 40 20 20 : 60
AA98 20 2B 40 40 40 40 40 2B : B6
AAA0 20 40 40 40 40 40 40 40 : E0
AAA8 20 2B 2B 2B 40 2B 2B 2B : 62
AAB0 20 20 20 40 40 40 20 20 : 60
AAB8 20 20 20 40 40 40 20 20 : 60
AAC0 20 20 20 20 20 20 20 20 : 00
AAC8 20 20 20 2B 2B 2B 20 20 : 21
AAD0 20 20 20 2B 40 2B 20 20 : 36
AAD8 20 2B 40 40 40 40 40 2B : B6
AAE0 20 2B 40 2B 40 2B 40 2B : 8C
AAE8 20 2B 40 40 40 40 40 2B : B6
AAF0 20 20 20 2B 40 2B 20 20 : 36
AAF8 20 20 20 2B 40 2B 20 20 : 36
-----------------------------------
SUM: 00 57 AB 2D AB 2D AB 57 507F

AB00 20 20 20 20 20 20 20 20 : 00
AB08 20 20 20 2B 2B 2B 20 20 : 21
AB10 20 20 20 40 40 40 20 20 : 60
AB18 20 2B 2B 40 2B 40 2B 2B : 77
AB20 20 2B 2B 40 40 40 2B 2B : 8C
AB28 20 2B 2B 40 2B 40 2B 2B : 77
AB30 20 20 20 40 40 40 20 20 : 60
AB38 20 20 20 2B 2B 2B 20 20 : 21
AB40 20 20 20 20 20 20 20 20 : 00
AB48 20 20 20 2B 2B 2B 20 20 : 21
AB50 20 20 20 2B 40 2B 20 20 : 36
AB58 20 40 40 40 40 40 40 40 : E0
AB60 20 2B 2B 40 40 40 2B 2B : 8C
AB68 20 2B 2B 2B 40 2B 2B 2B : 62
AB70 20 20 20 40 2B 40 20 20 : 4B
AB78 20 20 20 40 2B 40 20 20 : 4B
-----------------------------------
SUM: 00 57 57 57 2D 57 57 57 0188

AB80 20 20 20 20 20 20 20 20 : 00
AB88 20 20 20 40 2B 40 20 20 : 4B
AB90 20 20 20 2B 40 2B 20 20 : 36
AB98 20 40 40 40 40 40 40 40 : E0
ABA0 20 40 40 40 40 40 40 40 : E0
ABA8 20 2B 2B 2B 40 2B 2B 2B : 62
ABB0 20 20 20 2B 40 2B 20 20 : 36
ABB8 20 20 20 2B 40 2B 20 20 : 36
```

```
ABC0 20 20 20 20 20 20 20 20 : 00
ABC8 20 20 20 2B 2B 2B 20 20 : 21
ABD0 20 20 20 2B 40 2B 20 20 : 36
ABD8 20 2B 2B 40 2B 40 2B 2B : 77
ABE0 20 2B 40 40 40 40 40 2B : B6
ABE8 20 2B 2B 40 40 40 2B 2B : 8C
ABF0 20 20 20 2B 2B 2B 20 20 : 21
ABF8 20 20 20 2B 40 2B 20 20 : 36
-----------------------------------
SUM: 00 6C 81 18 6C 18 81 6C A01E

AC00 20 20 20 20 20 20 20 20 : 00
AC08 20 20 20 2B 2B 2B 20 20 : 21
AC10 20 20 20 2B 2B 2B 20 20 : 21
AC18 20 2B 40 40 40 40 40 2B : B6
AC20 20 2B 40 40 40 40 40 2B : B6
AC28 20 2B 40 40 40 40 40 2B : B6
AC30 20 20 20 40 40 40 20 20 : 60
AC38 20 20 20 2B 40 2B 20 20 : 36
AC40 20 20 20 20 20 20 20 20 : 00
AC48 20 20 20 40 2B 40 20 20 : 4B
AC50 20 20 20 2B 40 2B 20 20 : 36
AC58 20 40 40 40 40 40 40 40 : E0
AC60 20 40 40 40 40 40 40 40 : E0
AC68 20 2B 40 40 2B 40 40 2B : A1
AC70 20 20 20 40 2B 40 20 20 : 4B
AC78 20 20 20 40 2B 40 20 20 : 4B
-----------------------------------
SUM: 00 6C C0 6C 42 6C C0 6C 4D5F

AC80 20 20 20 20 20 20 20 20 : 00
AC88 20 20 20 2B 2B 2B 20 20 : 21
AC90 20 20 20 40 2B 40 20 20 : 4B
AC98 20 2B 2B 40 2B 40 2B 2B : 77
ACA0 20 2B 40 40 40 40 40 2B : B6
ACA8 20 2B 2B 40 40 40 2B 2B : 8C
ACB0 20 20 20 40 40 40 20 20 : 60
ACB8 20 20 20 2B 40 2B 20 20 : 36
ACC0 20 20 20 20 20 20 20 20 : 00
ACC8 20 20 20 2B 2B 2B 20 20 : 21
ACD0 20 20 20 2B 40 2B 20 20 : 36
ACD8 20 2B 40 2B 40 2B 40 2B : 8C
ACE0 20 2B 2B 40 40 40 2B 2B : 8C
ACE8 20 40 40 40 40 40 40 40 : E0
ACF0 20 20 20 40 40 40 20 20 : 60
ACF8 20 20 20 2B 40 2B 20 20 : 36
```

```
-----------------------------------
SUM: 00 57 81 42 6C 42 81 57 D96B

AD00 20 20 20 20 20 20 20 20 : 00
AD08 20 20 20 2B 40 2B 20 20 : 36
AD10 20 20 20 2B 40 2B 20 20 : 36
AD18 20 2B 2B 40 40 40 2B 2B : 8C
AD20 20 2B 40 40 40 40 40 2B : B6
AD28 20 40 2B 40 40 40 2B 40 : B6
AD30 20 20 20 2B 40 2B 20 20 : 36
AD38 20 20 20 2B 2B 2B 20 20 : 21
AD40 20 20 20 20 20 20 20 20 : 00
AD48 20 20 20 40 40 2B 20 20 : 4B
AD50 20 20 20 2B 40 40 20 20 : 4B
AD58 20 2B 2B 40 40 2B 2B 2B : 77
AD60 20 2B 40 40 2B 40 40 2B : A1
AD68 20 2B 2B 2B 40 40 2B 2B : 77
AD70 20 20 20 40 40 2B 20 20 : 4B
AD78 20 20 20 2B 40 40 20 20 : 4B
-----------------------------------
SUM: 00 57 6C 2D 96 2D 6C 57 EFD5

AD80 20 20 20 20 20 20 20 20 : 00
AD88 20 20 20 40 40 40 20 20 : 60
AD90 20 20 20 40 40 40 20 20 : 60
AD98 20 40 40 40 40 40 40 40 : E0
ADA0 20 40 40 40 2B 40 40 40 : CB
ADA8 20 40 40 40 40 40 40 40 : E0
ADB0 20 20 20 40 40 40 20 20 : 60
ADB8 20 20 20 40 40 40 20 20 : 60
ADC0 20 20 20 20 20 20 20 20 : 00
ADC8 20 20 20 2B 2B 2B 20 20 : 21
ADD0 20 20 20 2B 2B 2B 20 20 : 21
ADD8 20 2B 2B 2B 2B 2B 2B 2B : 4D
ADE0 20 2B 2B 2B 2B 2B 2B 2B : 4D
ADE8 20 2B 2B 2B 2B 2B 2B 2B : 4D
ADF0 20 20 20 2B 2B 2B 20 20 : 21
ADF8 20 20 20 2B 2B 2B 20 20 : 21
-----------------------------------
SUM: 00 81 81 2D 18 2D 81 81 F059

AE00 20 20 20 20 20 20 20 20 : 00
AE08 20                      : 20
-----------------------------------
SUM: 40 20 20 20 20 20 20 20 E066
```

リスト2 LAST ONEソースリスト

```
0000                      1 ;===================
0000                      2 ; LAST ONE for S-OS
0000                      3 ;
0000                      4 ;      by Naxima
0000                      5 ;===================
0000                      6
0000                      7 #GETPC EQU  $1F80
0000                      8 #PEEK@ EQU  $1F91
0000                      9 #POKE@ EQU  $1F97
0000                     10 #PRTHL EQU  $1FBE
0000                     11 #PRTHX EQU  $1FC1
0000                     12 #BELL  EQU  $1FC4
0000                     13 #GETKY EQU  $1FD0
0000                     14 #LPTOF EQU  $1FD6
0000                     15 #MPRNT EQU  $1FE2
0000                     16 #PRNTS EQU  $1FF1
0000                     17 #PRINT EQU  $1FF4
0000                     18 #VER   EQU  $1FF7
0000                     19 #COLD EQU   $1FFD
0000                     20 #CSR   EQU  $2018
0000                     21 #SCRN EQU   $201B
0000                     22 #LOC   EQU  $201E
0000                     23 #FLGET EQU  $2021
0000                     24 #WIDCH EQU  $2030
0000                     25 ca     EQU  $2039
0000                     26 jp     EQU  $203C
0000                     27 ld     EQU  $203F
0000                     28 PATNO EQU   0      ;IX+0
0000                     29 PATLM EQU   1      ;IX+1
0000                     30 LEFT  EQU   2      ;IX+2
0000                     31 PMAX  EQU   99
0000                     32 CNT   EQU   256
0000                     33
A000                     34          ORG   $A000
A000                     35
A000                     36 s
A000 CD F7 1F            37          CALL  #VER
A003 7D                  38          LD    A,L
A004 FE 20               39          CP    20H
A006 C2 C4 1F            40          JP    NZ,#BELL
A009 21 39 20            41          LD    HL,ca
A00C 11 00 00            42          LD    DE,0
A00F 01 09 00            43          LD    BC,9
A012 CD 97 1F            44          CALL  #POKE@
A015 CD 80 1F            45          CALL  #GETPC
A018                     46 PC
A018 11 E8 FF            47          LD    DE,s-PC
A01B 19                  48          ADD   HL,DE
A01C E5                  49          PUSH  HL
A01D DD E1               50          POP   IX
A01F 11 89 05            51          LD    DE,CALLR-s
A022 DD 19               52          ADD   IX,DE
A024 DD 22 3A 20         53          LD    (ca+1),IX
A028 DD 75 09            54          LD    (IX+CR0-CALLR+1),L
A02B DD 74 0A            55          LD    (IX+CR0-CALLR+2),H
A02E 11 10 00            56          LD    DE,JUMPR-CALLR
A031 DD 19               57          ADD   IX,DE
A033 DD 22 3D 20         58          LD    (jp+1),IX
A037 DD 75 07            59          LD    (IX+JR0-JUMPR+1),L
A03A DD 74 08            60          LD    (IX+JR0-JUMPR+2),H
A03D 11 0E 00            61          LD    DE,LDDER-JUMPR
A040 DD 19               62          ADD   IX,DE
A042 DD 22 40 20         63          LD    (ld+1),IX
A046 DD 75 09            64          LD    (IX+LR0-LDDER+1),L
A049 DD 74 0A            65          LD    (IX+LR0-LDDER+2),H
A04C 3E C3               66          LD    A,$C3
A04E 32 39 20            67          LD    (ca),A
A051 32 3C 20            68          LD    (jp),A
A054 32 3F 20            69          LD    (ld),A
A057                     70
A057 CD D6 1F            71          CALL  #LPTOF
A05A CD 3F 20 B8 05      72          CALL  ld DW WORK-s
A05F D5                  73          PUSH  DE
A060 DD E1               74          POP   IX
A062 DD 36 00 00         75          LD    (IX+PATNO),0
A066                     76
A066                     77 ;
A066                     78 ; DEMONSTRATION
A066                     79 ;
A066 CD 39 20 5C 02      80          CALL  ca DW SCREN1-s
A06B CD 39 20 6C 03      81          CALL  ca DW SCREN2-s
A070 21 13 11            82          LD    HL,$1113
A073 CD 1E 20            83          CALL  #LOC
A076 CD E2 1F            84          CALL  #MPRNT
A079 44 45 4D 4F 4E 53 54 85         DM    "DEMONSTRATION" DB 0
A080 52 41 54 49 4F 4E 00
A087 21 13 13            86          LD    HL,$1313
A08A CD 1E 20            87          CALL  #LOC
A08D CD E2 1F            88          CALL  #MPRNT
A090 50 52 45 53 53 20 5B 89         DM    "PRESS [S] KEY" DB 0
A097 53 5D 20 4B 45 59 00
A09E CD 3F 20 B5 00      90          CALL  ld DW DEMOK-s
A0A3 2A 22 20            91          LD    HL,(#FLGET+1)
A0A6 E5                  92          PUSH  HL
A0A7 ED 53 22 20         93          LD    (#FLGET+1),DE
A0AB CD 3F 20 BB 05      94          CALL  ld DW DEMOKD-s
A0B0 D5                  95          PUSH  DE
A0B1 FD E1               96          POP   IY
A0B3 18 28               97          JR    SEL
A0B5                     98
A0B5                     99 DEMOK
A0B5 CD 39 20 49 05     100          CALL  ca DW FLASH-s
A0BA FE 53              101          CP    "S"
A0BC 28 11              102          JR    Z,DEND
A0BE                    103 DK0
A0BE FD 7E 00           104          LD    A,(IY)
A0C1 FD 23              105          INC   IY
A0C3 B7                 106          OR    A
A0C4 C0                 107          RET   NZ
A0C5 CD 3F 20 BB 05     108          CALL  ld DW DEMOKD-s
A0CA D5                 109          PUSH  DE
A0CB FD E1              110          POP   IY
A0CD 18 EF              111          JR    DK0
A0CF                    112 ;
A0CF                    113 ; DEMO END
A0CF                    114 ;
A0CF                    115 DEND
A0CF E1                 116          POP   HL          ;DUMMY
A0D0 E1                 117          POP   HL          ;DUMMY
A0D1 E1                 118          POP   HL
A0D2 22 22 20           119          LD    (#FLGET+1),HL
A0D5 CD 39 20 5C 02     120          CALL  ca DW SCREN1-s
A0DA DD 34 00           121          INC   (IX+PATNO)
A0DD                    122
A0DD                    123 ;
A0DD                    124 ; SELECT PATTERN
A0DD                    125 ;
A0DD                    126 SEL
A0DD CD 39 20 6C 03     127          CALL  ca DW SCREN2-s
A0E2 30 35              128          JR    NC,EDIT
A0E4                    129 SL0
A0E4 CD 39 20 FC 03     130          CALL  ca DW LOCCB-s
A0E9 CD 39 20 74 05     131          CALL  ca DW FLGET-s
A0EE FE 2B              132          CP    "+"
A0F0 28 0A              133          JR    Z,PNINC
A0F2 FE 2D              134          CP    "-"
A0F4 28 14              135          JR    Z,PNDEC
A0F6 FE 0D              136          CP    $0D
A0F8 28 69              137          JR    Z,NORM
A0FA 18 E8              138          JR    SL0
A0FC                    139
A0FC                    140 ;
A0FC                    141 ;
A0FC                    142 PNINC
A0FC DD 7E 01           143          LD    A,(IX+PATLM)
A0FF DD 34 00           144          INC   (IX+PATNO)
A102 DD BE 00           145          CP    (IX+PATNO)
A105 30 D6              146          JR    NC,SEL
A107 AF                 147          XOR   A
A108 18 0A              148          JR    PD0
A10A                    149
A10A                    150 ;
A10A                    151 ;
A10A                    152 PNDEC
A10A DD 7E 00           153          LD    A,(IX+PATNO)
A10D D6 01              154          SUB   1
```

▶サンダーフォースIIスゲェー。ゲームセンターにあってもおかしくネェような画面じゃん（テレビで見たんだけど）。ちくしょー，X68000さえ持ってれば……。しょうがないからX1テープ版が出るまで待つか。　植木　繁吉　(18)　東京都

```
A10F 30 03              155             JR    NC,PD0
A111 DD 7E 01           156             LD    A,(IX+PATLM)
A114                    157 PD0
A114 DD 77 00           158             LD    (IX+PATNO),A
A117 18 C4              159             JR    SEL
A119                    160
A119                    161 ;
A119                    162 ; EDITABLE PATTERN
A119                    163 ;
A119                    164 EDIT
A119 CD 39 20 E9 03     165             CALL  ca DW PDENT-s
A11E                    166 ED0
A11E CD 39 20 FC 03     167             CALL  ca DW LOCCB-s
A123 CD 39 20 74 05     168             CALL  ca DW FLGET-s
A128 FE 2B              169             CP    "+"
A12A 28 D0              170             JR    Z,PNINC
A12C FE 2D              171             CP    "-"
A12E 28 DA              172             JR    Z,PNDEC
A130 FE 0D              173             CP    $0D
A132 28 2A              174             JR    Z,NORM'
A134 FE 20              175             CP    " "
A136 28 10              176             JR    Z,ED2
A138 FE 43              177             CP    "C"
A13A 20 05              178             JR    NZ,ED1
A13C CD 3C 20 0B 05     179             CALL  jp DW CLRPAT-s
A141                    180 ED1
A141 CD 39 20 13 04     181             CALL  ca DW MOVCHK-s
A146 18 D6              182             JR    ED0
A148                    183 ED2
A148 CD 1B 20           184             CALL  #SCRN
A14B EE 6B              185             XOR   $6B
A14D CD F4 1F           186             CALL  #PRINT
A150 CD C4 1F           187             CALL  #BELL
A153 CD 39 20 07 04     188             CALL  ca DW IADR-s
A158 7E                 189             LD    A,(HL)
A159 EE 6B              190             XOR   $6B
A15B 77                 191             LD    (HL),A
A15C 18 C0              192             JR    ED0
A15E                    193
A15E                    194 ;
A15E                    195 ; NORMAL MODE
A15E                    196 ;
A15E                    197 NORM'
A15E CD 39 20 EE 04     198             CALL  ca DW ADDNEW-s
A163                    199 NORM
A163 CD 39 20 91 03     200             CALL  ca DW SCREN2'-s
A168 CD 39 20 DD 04     201             CALL  ca DW BEEP-s
A16D                    202 NR0
A16D CD 39 20 FC 03     203             CALL  ca DW LOCCB-s
A172 CD 39 20 74 05     204             CALL  ca DW FLGET-s
A177 FE 2B              205             CP    "+"
A179 28 81              206             JR    Z,PNINC
A17B FE 2D              207             CP    "-"
A17D 28 8B              208             JR    Z,PNDEC
A17F FE 0D              209             CP    $0D
A181 28 E0              210             JR    Z,NORM
A183 FE 20              211             CP    " "
A185 28 07              212             JR    Z,NR1
A187 CD 39 20 13 04     213             CALL  ca DW MOVCHK-s
A18C 18 DF              214             JR    NR0
A18E                    215
A18E                    216 NR1
A18E CD 1B 20           217             CALL  #SCRN
A191 FE 40              218             CP    "@"
A193 28 08              219             JR    Z,NR3
A195 CD C4 1F           220             CALL  #BELL
A198                    221 NR2
A198 CD C4 1F           222             CALL  #BELL
A19B 18 D0              223             JR    NR0
A19D                    224
A19D                    225 NR3
A19D 3E 7B              226             LD    A,"■"
A19F CD F4 1F           227             CALL  #PRINT
A1A2                    228 NR4
A1A2 CD C4 1F           229             CALL  #BELL
A1A5 CD 39 20 FC 03     230             CALL  ca DW LOCCB-s
A1AA CD 39 20 74 05     231             CALL  ca DW FLGET-s
A1AF FE 20              232             CP    " "
A1B1 20 07              233             JR    NZ,NR5
A1B3 3E 40              234             LD    A,"@"
A1B5 CD F4 1F           235             CALL  #PRINT
A1B8 18 DE              236             JR    NR2
A1BA                    237
A1BA                    238 NR5
A1BA C5                 239             PUSH  BC
A1BB CD 39 20 62 04     240             CALL  ca DW JMPCHK-s
A1C0 30 06              241             JR    NC,NR6
A1C2 C1                 242             POP   BC
A1C3 CD C4 1F           243             CALL  #BELL
A1C6 18 DA              244             JR    NR4
A1C8                    245
A1C8                    246 NR6
A1C8 E1                 247             POP   HL            ;HL=BC
A1C9 C5                 248             PUSH  BC
A1CA 44                 249             LD    B,H
A1CB 4D                 250             LD    C,L
A1CC CD 39 20 FC 03     251             CALL  ca DW LOCCB-s
A1D1 3E 2B              252             LD    A,"+"
A1D3 CD F4 1F           253             CALL  #PRINT
A1D6 EB                 254             EX    DE,HL
A1D7 CD 1E 20           255             CALL  #LOC
A1DA 3E 2B              256             LD    A,"+"
A1DC CD F4 1F           257             CALL  #PRINT
A1DF C1                 258             POP   BC
A1E0 CD 39 20 FC 03     259             CALL  ca DW LOCCB-s
A1E5 3E 40              260             LD    A,"@"
A1E7 CD F4 1F           261             CALL  #PRINT
A1EA CD C4 1F           262             CALL  #BELL
A1ED                    263
A1ED                    264 ;
A1ED                    265 ; LAST ONE OR NOT
A1ED                    266 ;
A1ED                    267 LSTONE
A1ED C5                 268             PUSH  BC
A1EE 16 00              269             LD    D,0
A1F0 0E 07              270             LD    C,7
A1F2                    271 LO0
A1F2 06 07              272             LD    B,7
A1F4                    273 LO1
A1F4 CD 39 20 FC 03     274             CALL  ca DW LOCCB-s
A1F9 CD 1B 20           275             CALL  #SCRN
A1FC FE 40              276             CP    "@"
A1FE 20 01              277             JR    NZ,LO2
A200 14                 278             INC   D
A201                    279 LO2
A201 10 F1              280             DJNZ  LO1
A203 0D                 281             DEC   C
A204 20 EC              282             JR    NZ,LO0
A206 C1                 283             POP   BC
A207 15                 284             DEC   D
A208 28 05              285             JR    Z,LO3
A20A CD 3C 20 6D 01     286             CALL  jp DW NR0-s
A20F                    287
A20F                    288 LO3
A20F 21 0C 0C           289             LD    HL,$0C0C
A212 CD 1B 20           290             CALL  #SCRN
A215 47                 291             LD    B,A
A216 21 09 17           292             LD    HL,$1709
A219 CD 1E 20           293             CALL  #LOC
A21C 78                 294             LD    A,B
A21D FE 40              295             CP    "@"
A21F 20 18              296             JR    NZ,LO4
```

```
A221 CD E2 1F           297             CALL  #MPRNT
A224 43 4F 4E 47 52 41 54  298          DM    "CONGRATULATIONS!! " DB 0
A22B 55 4C 41 54 49 4F 4E
A232 53 21 21 20 00
A237 18 10              299             JR    LO5
A239                    300
A239                    301 LO4
A239 CD E2 1F           302             CALL  #MPRNT
A23C 52 45 47 52 45 54 46  303          DM    "REGRETFUL!! " DB 0
A243 55 4C 21 21 20 00
A249                    304 LO5
A249 CD 39 20 DD 04     305             CALL  ca DW BEEP-s
A24E                    306 LO6
A24E CD 39 20 74 05     307             CALL  ca DW FLGET-s
A253 FE 0D              308             CP    $0D
A255 20 F7              309             JR    NZ,LO6
A257 CD 3C 20 DD 00     310             CALL  jp DW SEL-s
A25C                    311
A25C                    312 ;
A25C                    313 ; INITALIZE SCREEN 1
A25C                    314 ;
A25C                    315 SCREN1
A25C AF                 316             XOR   A
A25D CD 30 20           317             CALL  #WIDCH
A260 21 06 01           318             LD    HL,$0106
A263 CD 1E 20           319             CALL  #LOC
A266 CD E2 1F           320             CALL  #MPRNT
A269 50 41 54 54 45 52 4E  321          DM    "PATTERN:" DB 0
A270 3A 00
A272 21 1A 01           322             LD    HL,$011A
A275 CD 1E 20           323             CALL  #LOC
A278 CD E2 1F           324             CALL  #MPRNT
A27B 3D 3D 3D 3D 3D 3D 3D  325          DM    "==========" DB 0
A282 3D 3D 3D 00
A286 21 1B 02           326             LD    HL,$021B
A289 CD 1E 20           327             CALL  #LOC
A28C CD E2 1F           328             CALL  #MPRNT
A28F 4C 41 53 54 20 4F 4E  329          DM    "LAST ONE" DB 0
A296 45 00
A298 21 1A 03           330             LD    HL,$031A
A29B CD 1E 20           331             CALL  #LOC
A29E CD E2 1F           332             CALL  #MPRNT
A2A1 3D 3D 3D 3D 3D 3D 3D  333          DM    "==========" DB 0
A2A8 3D 3D 3D 00
A2AC 21 1A 05           334             LD    HL,$051A
A2AF CD 1E 20           335             CALL  #LOC
A2B2 CD E2 1F           336             CALL  #MPRNT
A2B5 2B 2D 3A 53 45 4C 45  337          DM    "+-:SELECT" DB 0
A2BC 43 54 00
A2BF 21 1A 07           338             LD    HL,$071A
A2C2 CD 1E 20           339             CALL  #LOC
A2C5 CD E2 1F           340             CALL  #MPRNT
A2C8 43 52 3A 53 54 41 52  341          DM    "CR:START" DB 0
A2CF 54 00
A2D1 21 1A 09           342             LD    HL,$091A
A2D4 CD 1E 20           343             CALL  #LOC
A2D7 CD E2 1F           344             CALL  #MPRNT
A2DA 32 34 36 38 3A 4D 4F  345          DM    "2468:MOVE" DB 0
A2E1 56 45 00
A2E4 21 1A 0B           346             LD    HL,$0B1A
A2E7 CD 1E 20           347             CALL  #LOC
A2EA CD E2 1F           348             CALL  #MPRNT
A2ED 53 50 3A 53 45 54 2F  349          DM    "SP:SET/RESET" DB 0
A2F4 52 45 53 45 54 00
A2FA                    350
A2FA 21 09 03           351             LD    HL,$0309
A2FD CD 39 20 23 03     352             CALL  ca DW ?BLCK-s
A302 21 03 09           353             LD    HL,$0903
A305 CD 39 20 23 03     354             CALL  ca DW ?BLCK-s
A30A 21 09 09           355             LD    HL,$0909
A30D CD 39 20 23 03     356             CALL  ca DW ?BLCK-s
A312 21 0F 09           357             LD    HL,$090F
A315 CD 39 20 23 03     358             CALL  ca DW ?BLCK-s
A31A 21 09 0F           359             LD    HL,$0F09
A31D CD 39 20 23 03     360             CALL  ca DW ?BLCK-s
A322 C9                 361             RET
A323                    362
A323                    363 ?BLCK
A323 CD 3F 20 3B 03     364             CALL  ld DW ?B2-s
A328 0E 07              365             LD    C,7
A32A                    366 ?B0
A32A CD 1E 20           367             CALL  #LOC
A32D 06 07              368             LD    B,7
A32F                    369 ?B1
A32F 1A                 370             LD    A,(DE)
A330 13                 371             INC   DE
A331 CD F4 1F           372             CALL  #PRINT
A334 10 F9              373             DJNZ  ?B1
A336 24                 374             INC   H
A337 0D                 375             DEC   C
A338 20 F0              376             JR    NZ,?B0
A33A C9                 377             RET
A33B                    378
A33B                    379 ?B2
A33B 2B 2D 2D 2B 2D 2D 2B  380          DM    "+--+--+"
A342 21 20 20 21 20 20 21  381          DM    "!  !  !"
A349 21 20 20 21 20 20 21  382          DM    "!  !  !"
A350 2B 2D 2D 2B 2D 2D 2B  383          DM    "+--+--+"
A357 21 20 20 21 20 20 21  384          DM    "!  !  !"
A35E 21 20 20 21 20 20 21  385          DM    "!  !  !"
A365 2B 2D 2D 2B 2D 2D 2B  386          DM    "+--+--+"
A36C                    387
A36C                    388 ;
A36C                    389 ; INITIALIZE SCREEN 2
A36C                    390 ;
A36C                    391 SCREN2
A36C DD 7E 00           392             LD    A,(IX+PATNO)
A36F FE 1E              393             CP    30
A371 F5                 394             PUSH  AF
A372 38 1E              395             JR    C,SC20
A374 21 09 17           396             LD    HL,$1709
A377 CD 1E 20           397             CALL  #LOC
A37A CD E2 1F           398             CALL  #MPRNT
A37D 59 4F 55 20 43 41 4E  399          DM    "YOU CAN EDIT      " DB 0
A384 20 45 44 49 54 20 20
A38B 20 20 20 00
A38F 18 0E              400             JR    SC22          ;^^^^^
A391                    401
A391                    402 SCREN2'
A391 F5                 403             PUSH  AF
A392                    404 SC20
A392 21 09 17           405             LD    HL,$1709
A395 CD 1E 20           406             CALL  #LOC
A398 06 11              407             LD    B,17
A39A                    408 SC21
A39A CD F1 1F           409             CALL  #PRNTS
A39D 10 FB              410             DJNZ  SC21
A39F                    411 SC22
A39F CD 39 20 25 05     412             CALL  ca DW SVAREA-s
A3A4 21 0E 01           413             LD    HL,$010E
A3A7 CD 1E 20           414             CALL  #LOC
A3AA CD 39 20 E9 03     415             CALL  ca DW PDENT-s
A3AF                    416
A3AF DD 4E 00           417             LD    C,(IX+PATNO)
A3B2 06 08              418             LD    B,8
A3B4 AF                 419             XOR   A
A3B5                    420 SC23
A3B5 CB 21              421             SLA   C
A3B7 17                 422             RLA
A3B8 FE 0A              423             CP    10
A3BA 38 03              424             JR    C,SC24
A3BC 0C                 425             INC   C
```


▶J&Pに通うこと約1週間。ついにサンダーフォースIIを終わらせた。1日数回にとどめたのがよかったのだろう。しかし，5面の横スクロールは1週間もかかってしまった。さあ，明後日の入試もがんばろう（だめだな，こりゃ）。　中原　隆彰　(18)　和歌山県

```
A3BD D6 0A              426             SUB    10
A3BF                    427 SC24
A3BF 10 F4              428             DJNZ   SC23         ;No./10=C...A
A3C1 CB 21              429             SLA    C
A3C3 CB 21              430             SLA    C
A3C5 CB 21              431             SLA    C
A3C7 CB 21              432             SLA    C
A3C9 B1                 433             OR     C            ;Cx16+A
A3CA CD C1 1F           434             CALL   #PRTHX
A3CD                    435
A3CD 0E 07              436             LD     C,7          ;X
A3CF                    437 SC25
A3CF 06 07              438             LD     B,7          ;Y
A3D1                    439 SC26
A3D1 CD 39 20 FC 03     440             CALL   ca DW LOCCB-s
A3D6 CD 39 20 07 04     441             CALL   ca DW IADR-s
A3DB 7E                 442             LD     A,(HL)
A3DC CD F4 1F           443             CALL   #PRINT
A3DF 10 F0              444             DJNZ   SC26
A3E1 0D                 445             DEC    C
A3E2 20 EB              446             JR     NZ,SC25
A3E4 01 01 03           447             LD     BC,$0301
A3E7 F1                 448             POP    AF
A3E8 C9                 449             RET
A3E9                    450
A3E9                    451 ;
A3E9                    452 ; PATTERN NO. x 64 + PATDT
A3E9                    453 ;
A3E9                    454 PDENT
A3E9 DD 6E 00           455             LD     L,(IX+PATNO)
A3EC                    456 PDENT'
A3EC 26 00              457             LD     H,0
A3EE 29                 458             ADD    HL,HL
A3EF 29                 459             ADD    HL,HL
A3F0 29                 460             ADD    HL,HL
A3F1 29                 461             ADD    HL,HL
A3F2 29                 462             ADD    HL,HL
A3F3 29                 463             ADD    HL,HL        ;x64
A3F4 CD 3F 20 49 06     464             CALL   ld DW PATDT-s
A3F9 19                 465             ADD    HL,DE
A3FA EB                 466             EX     DE,HL
A3FB C9                 467             RET
A3FC                    468
A3FC                    469 ;
A3FC                    470 ; LOCATE C,B
A3FC                    471 ;
A3FC                    472 LOCCB
A3FC 78                 473             LD     A,B
A3FD 87                 474             ADD    A,A
A3FE 80                 475             ADD    A,B
A3FF 67                 476             LD     H,A
A400 79                 477             LD     A,C
A401 87                 478             ADD    A,A
A402 81                 479             ADD    A,C
A403 6F                 480             LD     L,A
A404 C3 1E 20           481             JP     #LOC
A407                    482
A407                    483 ;
A407                    484 ; HL=DE+(B-1)x8+(Y-1)
A407                    485 ;
A407                    486 IADR
A407 78                 487             LD     A,B
A408 87                 488             ADD    A,A
A409 87                 489             ADD    A,A
A40A 87                 490             ADD    A,A
A40B 81                 491             ADD    A,C
A40C D6 09              492             SUB    9
A40E 26 00              493             LD     H,0
A410 6F                 494             LD     L,A
A411 19                 495             ADD    HL,DE
A412 C9                 496             RET
A413                    497
A413                    498 ;
A413                    499 ; MOVEMENT CHECK
A413                    500 ;
A413                    501 MOVCHK
A413 FE 32              502             CP     "2"
A415 28 22              503             JR     Z,MC2
A417 FE 34              504             CP     "4"
A419 28 27              505             JR     Z,MC4
A41B FE 36              506             CP     "6"
A41D 28 2A              507             JR     Z,MC6
A41F FE 38              508             CP     "8"
A421 28 0F              509             JR     Z,MC8
A423 FE 1F              510             CP     $1F
A425 28 12              511             JR     Z,MC2
A427 FE 1D              512             CP     $1D
A429 28 17              513             JR     Z,MC4
A42B FE 1C              514             CP     $1C
A42D 28 1A              515             JR     Z,MC6
A42F FE 1E              516             CP     $1E
A431 C0                 517             RET    NZ
A432                    518 MC8
A432 05                 519             DEC    B
A433 CD 39 20 52 04     520             CALL   ca DW MOVSUB-s
A438 D0                 521             RET    NC
A439                    522 MC2
A439 04                 523             INC    B
A43A CD 39 20 52 04     524             CALL   ca DW MOVSUB-s
A43F D0                 525             RET    NC
A440 05                 526             DEC    B
A441 C9                 527             RET
A442                    528
A442                    529 MC4
A442 0D                 530             DEC    C
A443 CD 39 20 52 04     531             CALL   ca DW MOVSUB-s
A448 D0                 532             RET    NC
A449                    533 MC6
A449 0C                 534             INC    C
A44A CD 39 20 52 04     535             CALL   ca DW MOVSUB-s
A44F D0                 536             RET    NC
A450 0D                 537             DEC    C
A451 C9                 538             RET
A452                    539 ;
A452                    540 MOVSUB
A452 CD 39 20 FC 03     541             CALL   ca DW LOCCB-s
A457 CD 1B 20           542             CALL   #SCRN
A45A FE 2B              543             CP     "+"
A45C C8                 544             RET    Z
A45D FE 40              545             CP     "@"
A45F C8                 546             RET    Z
A460 37                 547             SCF
A461 C9                 548             RET
A462                    549
A462                    550 ;
A462                    551 ; JUMP CHECK
A462                    552 ;
A462                    553 JMPCHK
A462 FE 32              554             CP     "2"
A464 28 1E              555             JR     Z,JC2
A466 FE 34              556             CP     "4"
A468 28 24              557             JR     Z,JC4
A46A FE 36              558             CP     "6"
A46C 28 2A              559             JR     Z,JC6
A46E FE 38              560             CP     "8"
A470 28 30              561             JR     Z,JC8
A472 FE 1F              562             CP     $1F
A474 28 0E              563             JR     Z,JC2
A476 FE 1D              564             CP     $1D
A478 28 14              565             JR     Z,JC4
A47A FE 1C              566             CP     $1C
A47C 28 1A              567             JR     Z,JC6
A47E FE 1E              568             CP     $1E
A480 28 20              569             JR     Z,JC8
A482 37                 570             SCF
A483 C9                 571             RET
A484                    572
A484                    573 JC2
A484 04                 574             INC    B
A485 CD 39 20 AC 04     575             CALL   ca DW JPSUB1-s
A48A D8                 576             RET    C
A48B 04                 577             INC    B
A48C 18 2D              578             JR     JPSUB2
A48E                    579
A48E                    580 JC4
A48E 0D                 581             DEC    C
A48F CD 39 20 AC 04     582             CALL   ca DW JPSUB1-s
A494 D8                 583             RET    C
A495 0D                 584             DEC    C
A496 18 23              585             JR     JPSUB2
A498                    586
A498                    587 JC6
A498 0C                 588             INC    C
A499 CD 39 20 AC 04     589             CALL   ca DW JPSUB1-s
A49E D8                 590             RET    C
A49F 0C                 591             INC    C
A4A0 18 19              592             JR     JPSUB2
A4A2                    593
A4A2                    594 JC8
A4A2 05                 595             DEC    B
A4A3 CD 39 20 AC 04     596             CALL   ca DW JPSUB1-s
A4A8 D8                 597             RET    C
A4A9 05                 598             DEC    B
A4AA 18 0F              599             JR     JPSUB2
A4AC                    600
A4AC                    601 ;
A4AC                    602 ;
A4AC                    603 JPSUB1
A4AC CD 39 20 FC 03     604             CALL   ca DW LOCCB-s
A4B1 54                 605             LD     D,H
A4B2 5D                 606             LD     E,L
A4B3 CD 1B 20           607             CALL   #SCRN
A4B6 FE 40              608             CP     "@"
A4B8 C8                 609             RET    Z
A4B9 37                 610             SCF
A4BA C9                 611             RET
A4BB                    612
A4BB                    613 ;
A4BB                    614 ;
A4BB                    615 JPSUB2
A4BB CD 39 20 FC 03     616             CALL   ca DW LOCCB-s
A4C0 CD 1B 20           617             CALL   #SCRN
A4C3 FE 2B              618             CP     "+"
A4C5 C8                 619             RET    Z
A4C6 37                 620             SCF
A4C7 C9                 621             RET
A4C8                    622
A4C8                    623 ;
A4C8                    624 ; WAIT
A4C8                    625 ;
A4C8                    626 WAIT1
A4C8 CD 39 20 CD 04     627             CALL   ca DW WAIT2-10-s
A4CD CD 39 20 D2 04     628             CALL   ca DW WAIT2-5-s
A4D2 CD 39 20 D7 04     629             CALL   ca DW WAIT2-s
A4D7                    630 WAIT2
A4D7 3E 00              631             LD     A,CNT
A4D9                    632 WT0
A4D9 3D                 633             DEC    A
A4DA 20 FD              634             JR     NZ,WT0
A4DC C9                 635             RET
A4DD                    636
A4DD                    637 ;
A4DD                    638 ; BEEP
A4DD                    639 ;
A4DD                    640 BEEP
A4DD C5                 641             PUSH   BC
A4DE 06 04              642             LD     B,4
A4E0                    643 BP0
A4E0 CD C4 1F           644             CALL   #BELL
A4E3 CD 39 20 C8 04     645             CALL   ca DW WAIT1-s
A4E8 10 F6              646             DJNZ   BP0
A4EA C1                 647             POP    BC
A4EB C3 C4 1F           648             JP     #BELL
A4EE                    649
A4EE                    650 ;
A4EE                    651 ; ADD NEW PATTERN
A4EE                    652 ;
A4EE                    653 ADDNEW
A4EE DD 7E 01           654             LD     A,(IX+PATLM)
A4F1 DD BE 00           655             CP     (IX+PATNO)
A4F4 C0                 656             RET    NZ
A4F5 FE 63              657             CP     PMAX
A4F7 C8                 658             RET    Z
A4F8 DD 34 01           659             INC    (IX+PATLM)
A4FB CD 39 20 E9 03     660             CALL   ca DW PDENT-s
A500 01 40 00           661             LD     BC,64
A503 EB                 662             EX     DE,HL
A504 09                 663             ADD    HL,BC
A505 EB                 664             EX     DE,HL
A506 ED 42              665             SBC    HL,BC        ;CF=0, HL=PATDT
A508 ED B0              666             LDIR
A50A C9                 667             RET
A50B                    668
A50B                    669 ;
A50B                    670 ; CLEAR USER'S PATTERN
A50B                    671 ;
A50B                    672 CLRPAT
A50B CD C4 1F           673             CALL   #BELL
A50E DD 7E 00           674             LD     A,(IX+PATNO)
A511 3D                 675             DEC    A
A512 DD 77 00           676             LD     (IX+PATNO),A
A515 DD 77 01           677             LD     (IX+PATLM),A
A518 CD 39 20 EE 04     678             CALL   ca DW ADDNEW-s
A51D DD 34 00           679             INC    (IX+PATNO)
A520 CD 3C 20 DD 00     680             CALL   jp DW SEL-s
A525                    681
A525                    682 ;
A525                    683 ; PROGRAM AREA
A525                    684 ;
A525                    685 SVAREA
A525 21 1A 0D           686             LD     HL,$0D1A
A528 CD 1E 20           687             CALL   #LOC
A52B CD 3F 20 00 00     688             CALL   ld DW 0
A530 EB                 689             EX     DE,HL
A531 CD BE 1F           690             CALL   #PRTHL
A534 CD E2 1F           691             CALL   #MPRNT
A537 20 2D 20 00        692             DM     " - " DB 0
A53B DD 6E 01           693             LD     L,(IX+PATLM)
A53E 2C                 694             INC    L
A53F CD 39 20 EC 03     695             CALL   ca DW PDENT'-s
A544 1B                 696             DEC    DE
A545 EB                 697             EX     DE,HL
A546 C3 BE 1F           698             JP     #PRTHL
A549                    699
A549                    700 ;
A549                    701 ; FLASH
A549                    702 ;
A549                    703 FLASH
A549 C5                 704             PUSH   BC
A54A D5                 705             PUSH   DE
A54B E5                 706             PUSH   HL
A54C CD 18 20           707             CALL   #CSR
A54F CD 1B 20           708             CALL   #SCRN
A552 57                 709             LD     D,A
```


▶ドラゴンスピリットのオープニングを見る限り，ブルードラゴンには前足がなく，翼に爪が生えている。これではどう考えてもドラゴンではなく，ワイバーンである。だから本当は「ワイバーンスピリット」が正解である。　高山　稔　(19)　神奈川県

```
A553 01 02 00         710             LD    BC,2
A556 3E 7B            711             LD    A,"■"
A558                  712 FL0
A558 CD F4 1F         713             CALL  #PRINT
A55B CD 1E 20         714             CALL  #LOC
A55E                  715 FL1
A55E CD 39 20 D7 04   716             CALL  ca DW WAIT2-s
A563 CD D0 1F         717             CALL  #GETKY
A566 FE 53            718             CP    "S"
A568 28 06            719             JR    Z,FL2
A56A 10 F2            720             DJNZ  FL1
A56C 7A               721             LD    A,D
A56D 0D               722             DEC   C
A56E 20 E8            723             JR    NZ,FL0
A570                  724 FL2
A570 E1               725             POP   HL
A571 D1               726             POP   DE
A572 C1               727             POP   BC
A573 C9               728             RET
A574                  729
A574                  730 ;
A574                  731 ; #FLGET & INTERNMISSION
A574                  732 ;
A574                  733 FLGET
A574 CD 21 20         734             CALL  #FLGET
A577 FE 1B            735             CP    $1B
A579 C0               736             RET   NZ
A57A 21 39 20         737             LD    HL,ca
A57D 11 00 00         738             LD    DE,0
A580 01 09 00         739             LD    BC,9
A583 CD 91 1F         740             CALL  #PEEK@
A586 C3 FD 1F         741             JP    #COLD
A589                  742
A589                  743 ;
A589                  744 ; CALL RELATIVE
A589                  745 ;
A589                  746 CALLR
A589 08               747             EX    AF,AF'
A58A D9               748             EXX
A58B E1               749             POP   HL
A58C 5E               750             LD    E,(HL)
A58D 23               751             INC   HL
A58E 56               752             LD    D,(HL)
A58F 23               753             INC   HL
A590 E5               754             PUSH  HL
A591                  755 CR0
A591 21 00 00         756             LD    HL,0
A594 19               757             ADD   HL,DE
A595 E5               758             PUSH  HL
A596 D9               759             EXX
A597 08               760             EX    AF,AF'
A598 C9               761             RET
A599                  762
A599                  763 ;
A599                  764 ; JUMP RELATIVE
A599                  765 ;
A599                  766 JUMPR
A599 08               767             EX    AF,AF'
A59A D9               768             EXX
A59B E1               769             POP   HL
A59C 5E               770             LD    E,(HL)
A59D 23               771             INC   HL
A59E 56               772             LD    D,(HL)
A59F                  773 JR0
A59F 21 00 00         774             LD    HL,0
A5A2 19               775             ADD   HL,DE
A5A3 E5               776             PUSH  HL
A5A4 D9               777             EXX
A5A5 08               778             EX    AF,AF'
A5A6 C9               779             RET
A5A7                  780
A5A7                  781 ;
A5A7                  782 ; LOAD DE RELATIVE
A5A7                  783 ;
A5A7                  784 LDDER
A5A7 08               785             EX    AF,AF'
A5A8 D9               786             EXX
A5A9 E1               787             POP   HL
A5AA 5E               788             LD    E,(HL)
A5AB 23               789             INC   HL
A5AC 56               790             LD    D,(HL)
A5AD 23               791             INC   HL
A5AE E5               792             PUSH  HL
A5AF                  793 LR0
A5AF 21 00 00         794             LD    HL,0
A5B2 19               795             ADD   HL,DE
A5B3 E5               796             PUSH  HL
A5B4 D9               797             EXX
A5B5 08               798             EX    AF,AF'
A5B6 D1               799             POP   DE
A5B7 C9               800             RET
A5B8                  801
A5B8                  802 ;
A5B8                  803 ; WORK AREA
A5B8                  804 ;
A5B8                  805 WORK
A5B8 00               806             DB    0               ;PATNO
A5B9 1E               807             DB    30              ;PATLM
A5BA 00               808             DB    0               ;LEFT
A5BB                  809
A5BB                  810 ;
A5BB                  811 ; DATA AREA
A5BB                  812 ;
A5BB                  813 DEMOKD
A5BB 2A 0D            814             DB    $2A,$0D
A5BD 36 36 36 32 20 36 34  815        DM    "6662 64422 84 66 4***"
A5C4 34 32 32 20 38 34 20
A5CB 36 36 20 34 2A 2A 2A
A5D2 0D 00            816             DB    $0D,0
A5D4 00 00 00 00 00 00 00  817        DS    53
A5DB 00 00 00 00 00 00 00
A5E2 00 00 00 00 00 00 00
A5E9 00 00 00 00 00 00 00
A5F0 00 00 00 00 00 00 00
A5F7 00 00 00 00 00 00 00
A5FE 00 00 00 00 00 00 00
A605 00 00 00 00
```

リスト3　面データ

```
 817
 818  DM  "  +++   "  ;CLEAR
 819  DM  "  +++   "
 820  DM  "+++++++ "
 821  DM  "+++++++ "
 822  DM  "+++++++ "
 823  DM  "  +++   "
 824  DM  "  +++   "
 825  DM  "        "
 826
 827 PATDT
 828  DM  "  +++   "  ;00
 829  DM  "  +++   "
 830  DM  "+++++++ "
 831  DM  "++@@@++ "
 832  DM  "+++@+++ "
 833  DM  "  +@+   "
 834  DM  "  +++   "
 835  DM  "        "
 836
 837  DM  "  +++   "  ;01
 838  DM  "  +@+   "
 839  DM  "++@@@++ "
 840  DM  "+++@+++ "
 841  DM  "+++@+++ "
 842  DM  "  +++   "
 843  DM  "  +++   "
 844  DM  "        "
 845
 846  DM  "  +++   "  ;02
 847  DM  "  +++   "
 848  DM  "+++@+++ "
 849  DM  "++@@@++ "
 850  DM  "+@@@@@+ "
 851  DM  "  +++   "
 852  DM  "  +++   "
 853  DM  "        "
 854
 855  DM  "  +++   "  ;03
 856  DM  "  +++   "
 857  DM  "+@@+@@+ "
 858  DM  "+++@+++ "
 859  DM  "+++++++ "
 860  DM  "  +++   "
 861  DM  "  +++   "
 862  DM  "        "
 863
 864  DM  "  +++   "  ;04
 865  DM  "  +@+   "
 866  DM  "++@@@++ "
 867  DM  "+++@+++ "
 868  DM  "+@@@@@+ "
 869  DM  "  +++   "
 870  DM  "  +++   "
 871  DM  "        "
 872
 873  DM  "  @@@   "  ;05
 874  DM  "  @@@   "
 875  DM  "++@@@++ "
 876  DM  "+@@@@@+ "
 877  DM  "+++++++ "
 878  DM  "  +++   "
 879  DM  "  +++   "
 880  DM  "        "
 881
 882  DM  "  +++   "  ;06
 883  DM  "  +@+   "
 884  DM  "++@@@++ "
 885  DM  "+@@@@@+ "
 886  DM  "++@+@++ "
 887  DM  "  +++   "
 888  DM  "  +++   "
 889  DM  "        "
 890
 891  DM  "  @+@   "  ;07
 892  DM  "  @@@   "
 893  DM  "@@+@+@@ "
 894  DM  "+@@@@@+ "
 895  DM  "@@+@+@@ "
 896  DM  "  @@@   "
 897  DM  "  @+@   "
 898  DM  "        "
 899
 900  DM  "  +++   "  ;08
 901  DM  "  +@+   "
 902  DM  "++@@@++ "
 903  DM  "+@@+@@+ "
 904  DM  "++@@@++ "
 905  DM  "  +@+   "
 906  DM  "  +++   "
 907  DM  "        "
 908
 909  DM  "  +@+   "  ;09
 910  DM  "  @@@   "
 911  DM  "+@@@@@+ "
 912  DM  "@@@@@@@ "
 913  DM  "++@+@++ "
 914  DM  "  +++   "
 915  DM  "  +++   "
 916  DM  "        "
 917
 918  DM  "  +++   "  ;10
 919  DM  "  +++   "
 920  DM  "+++++++ "
 921  DM  "+++@+++ "
 922  DM  "++@@@++ "
 923  DM  "  @@@   "
 924  DM  "  @@@   "
 925  DM  "        "
 926
 927  DM  "  +++   "  ;11
 928  DM  "  +@+   "
 929  DM  "++@@@++ "
 930  DM  "+@@@@@+ "
 931  DM  "@@@@@@@ "
 932  DM  "  +++   "
 933  DM  "  +++   "
 934  DM  "        "
 935
 936  DM  "  +++   "  ;12
 937  DM  "  +@+   "
 938  DM  "++@@@++ "
 939  DM  "++@@@++ "
 940  DM  "++@@@++ "
 941  DM  "  +@+   "
 942  DM  "  +++   "
 943  DM  "        "
 944
 945  DM  "  +++   "  ;13
 946  DM  "  +++   "
 947  DM  "+++@+++ "
 948  DM  "+@@@@@+ "
 949  DM  "++@@@++ "
 950  DM  "  @+@   "
 951  DM  "  +++   "
 952  DM  "        "
 953
 954  DM  "  +@+   "  ;14
 955  DM  "  +@+   "
 956  DM  "+@@@@@+ "
 957  DM  "@@@@@@@ "
 958  DM  "+@@@@@+ "
 959  DM  "  +@+   "
 960  DM  "  +@+   "
 961  DM  "        "
 962
 963  DM  "  +++   "  ;15
 964  DM  "  +@+   "
 965  DM  "++@@@++ "
 966  DM  "@@@@@@@ "
 967  DM  "++@@@++ "
 968  DM  "  +@+   "
 969  DM  "  +@+   "
 970  DM  "        "
 971
 972  DM  "  +@+   "  ;16
 973  DM  "  @@@   "
 974  DM  "@@+@+@@ "
 975  DM  "+@@@@@+ "
 976  DM  "++@@@++ "
 977  DM  "  @@@   "
 978  DM  "  +@+   "
 979  DM  "        "
 980
 981  DM  "  +@+   "  ;17
 982  DM  "  @@@   "
 983  DM  "+@@@@@+ "
 984  DM  "@@@@@@@ "
 985  DM  "+++@+++ "
 986  DM  "  @@@   "
 987  DM  "  @@@   "
 988  DM  "        "
 989
 990  DM  "  +++   "  ;18
 991  DM  "  +@+   "
 992  DM  "+@@@@@+ "
 993  DM  "+@+@+@+ "
 994  DM  "+@@@@@+ "
 995  DM  "  +@+   "
 996  DM  "  +@+   "
 997  DM  "        "
 998
 999  DM  "  +++   "  ;19
1000  DM  "  @@@   "
1001  DM  "++@+@++ "
1002  DM  "++@@@++ "
1003  DM  "++@+@++ "
1004  DM  "  @@@   "
1005  DM  "  +++   "
1006  DM  "        "
1007
1008  DM  "  +++   "  ;20
1009  DM  "  +@+   "
1010  DM  "@@@@@@@ "
1011  DM  "++@@@++ "
1012  DM  "+++@+++ "
1013  DM  "  @+@   "
1014  DM  "  @+@   "
1015  DM  "        "
1016
1017  DM  "  @+@   "  ;21
1018  DM  "  +@+   "
1019  DM  "@@@@@@@ "
1020  DM  "@@@@@@@ "
1021  DM  "+++@+++ "
1022  DM  "  +@+   "
1023  DM  "  +@+   "
1024  DM  "        "
1025
1026  DM  "  +++   "  ;22
1027  DM  "  +@+   "
1028  DM  "++@+@++ "
1029  DM  "+@@@@@+ "
1030  DM  "++@@@++ "
1031  DM  "  +++   "
1032  DM  "  +@+   "
1033  DM  "        "
1034
1035  DM  "  +++   "  ;23
1036  DM  "  +++   "
1037  DM  "+@@@@@+ "
1038  DM  "+@@@@@+ "
1039  DM  "+@@@@@+ "
1040  DM  "  @@@   "
1041  DM  "  +@+   "
1042  DM  "        "
1043
1044  DM  "  @+@   "  ;24
1045  DM  "  +@+   "
1046  DM  "@@@@@@@ "
1047  DM  "@@@@@@@ "
1048  DM  "+@@+@@+ "
1049  DM  "  @+@   "
1050  DM  "  @+@   "
1051  DM  "        "
1052
1053  DM  "  +++   "  ;25
1054  DM  "  @+@   "
1055  DM  "++@+@++ "
1056  DM  "+@@@@@+ "
1057  DM  "++@@@++ "
1058  DM  "  @@@   "
1059  DM  "  +@+   "
1060  DM  "        "
1061
1062  DM  "  +++   "  ;26
1063  DM  "  +@+   "
1064  DM  "+@+@+@+ "
1065  DM  "++@@@++ "
1066  DM  "@@@@@@@ "
1067  DM  "  @@@   "
1068  DM  "  +@+   "
1069  DM  "        "
1070
1071  DM  "  +@+   "  ;27
1072  DM  "  +@+   "
1073  DM  "++@@@++ "
1074  DM  "+@@@@@+ "
1075  DM  "@+@@@+@ "
1076  DM  "  +@+   "
1077  DM  "  +++   "
1078  DM  "        "
1079
1080  DM  "  @@+   "  ;28
1081  DM  "  +@@   "
1082  DM  "++@@+++ "
1083  DM  "+@@+@@+ "
1084  DM  "+++@@++ "
1085  DM  "  @@+   "
1086  DM  "  +@@   "
1087  DM  "        "
1088
1089  DM  "  @@@   "  ;29
1090  DM  "  @@@   "
1091  DM  "@@@@@@@ "
1092  DM  "@@@+@@@ "
1093  DM  "@@@@@@@ "
1094  DM  "  @@@   "
1095  DM  "  @@@   "
1096  DM  "        "
1097
1098  DM  "  +++   "  ;30
1099  DM  "  +++   "
1100  DM  "+++++++ "
1101  DM  "+++++++ "
1102  DM  "+++++++ "
1103  DM  "  +++   "
1104  DM  "  +++   "
1105  DM  "        "
```


▶私はイースIIの神殿のマップを書いた。ちょっとは同情してちょーだいな。
宮田　順朗　(19)　埼玉県

全機種共通S-OS“SWORD”要

ブロックゲームFLICK

パズルゲーム第2弾はFLICK，SMCユーザーからの投稿です。数字の分だけブロックを移動できるという，ちょっと変わったルールに従ってキャラクターを動かしていきます。なかなか歯ごたえのあるユニークなパズルゲームです。

佐藤 義弘 *Sato Yoshihiro*

パズルをもうひとつ

S-OS“SWORD”上で動作するブロックタイプのパズルゲームです。リアルタイムではありませんから，じっくり考えても大丈夫。詰め将棋のつもりで挑戦してみてください。

入力の際は各機種のマシン語モニタからMコマンドなどのメモリチェンジ用コマンドを用いるか，またはMACINTO-Cなどのマシン語入力ツールを使ってリスト1を打ち込んでください。チェックサム，CRCチェックバイトなどをよく確認したうえで，

#S　FLICK : 5000 : 57FF : 5000

のようにセーブしてください。

実行の際はS-OSのモニタ上から5000Hにジャンプ（#J5000）してください。

基本的な操作は，キャラクターの移動がテンキーの2，4，6，8，カーソルキーの上下左右，フルキーのI，J，K，Lに対応しているほか，

R	やりなおし
N	次の面に進む
B	前の面に戻る
S	指定した面に進む。移動キーで面を選んでリターンキーで決定
Q	ゲームを終了し，S-OSのホットスタートに戻る

の各キーに機能が割り振られています。

ルール&ゲーム内容

◇のキャラクターを使ってʊʊを∷のところまで運んでいくことがこのパズルの目的です。ただし，ʊʊを直接押していくことはできません。もちろん，引っ張ることもできません。

それでは，どのようにして動かしていくのかといいますと，まず数字の書かれたブロックをʊʊの隣に運びます。この数字の書かれたブロックは◇を使って押していくことができます（引っ張ることはできない)。続いて動かしたい方向に押してやると，ブロックに書かれた数字分だけʊʊが移動していきます。また，途中でなにかにぶつかるとʊʊは停止します。当然まわりの壁などの障害物は動かすことができません。ブロックをうまく使って移動量をコントロールしながら，ʊʊを∷のところまで運んでいくわけです。起動してしばらくそのまま放っておくとタイトル表示のあとに1面のデモが始まります。それを見れば1面の解答がわかりますから，それで動かし方のコツをつかむとよいでしょう。

うまくゴールにたどり着いたら，次の面に進みます。途中で行き詰まった場合はRキーで最初からやりなおしてください。キャラクターを操作するたびにステップ数が記録されていきます。面によっては複数の解答が存在する場合もありえますので，できるだけ短いステップ数で解くようにしてみましょう。

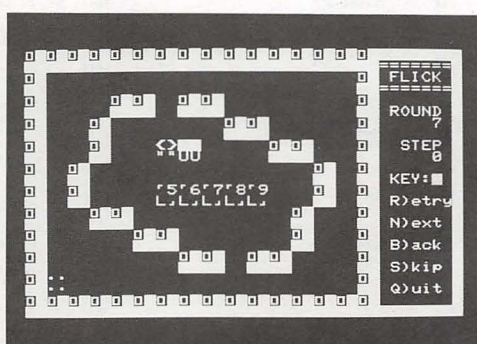
プログラムについて

ソースプログラムはオリジナルアセンブラで記述されています。マクロ命令などが使ってありZEDAでそのまま使うことはできませんので，参考程度に見ておいてください。

プログラム起動時のデモは機種によってかなりスピードに差があります。X1やSMCでデモのスピードが速すぎたり，MZ系の機種で遅すぎたりして気にいらないという場合は，5402Hのデータ部分（標準状態では30H）を書き換えるとよいでしょう。ここに格納されているデータが小さいほどスピードが速くなります（ただし，00HはFFH＋1とみなされるので注意)。

面データは全部で16面分用意してあります。5003Hのデータが最大面数を表していますので，自分で面を増やしたいときにはここのデータを書き換え，5800H以降にデータを置いてプログラムをセーブしなおしてください。

面データはひとつの面あたり32バイトで構成されています。その内訳は最初の20バイトが空白部分と壁のデータを2進数で表したもので，次の9バイトは1～9までの数字ブロックの位置を表し（置かないときは00H)，残りの3バイトで∷，ʊʊ，◇の初期位置を指定しています。詳しくは1～16面のデータを参照してください。

Profile
◇佐藤さんは東京にお住まいの20歳，大学2年生です。パソコン歴は約6年のSMC-777ユーザー。S-OSユーザーですがスクリーンエディタやアセンブラなどは自作して使用されているそうです。

リスト1 FLICKダンプリスト

```
5000 C3 7D 53 10 20 20 20 20 : 23
5008 A2 20 4C A3 7B 7B 55 55 : 51
5010 A5 A5 A5 A5 3C 3E 22 22 : 52
5018 DB 7B 7B 7B 3E AF 32 4A : B5
5020 43 3E 0C CD F4 1F CD C5 : FF
5028 52 3E 01 32 4B 43 31 00 : 82
5030 43 CD 22 52 CD 47 53 CD : B8
5038 6E 50 30 05 CD 7D 50 18 : A5
5040 F3 CD D1 50 3A 49 43 B7 : 5E
5048 28 EA 11 3D 55 CD E8 52 : BC
5050 3A 4A 43 B7 20 12 CD CA : 47
5058 1F FE 1B CA 68 53 FE 20 : DB
5060 20 F4 CD 8E 50 C3 88 53 : 5D
5068 CD EF 53 C3 8E 53 21 57 : 2B
5070 55 01 0C 00 ED B1 37 C0 : F7
5078 79 E6 03 4F C9 4F 7E 23 : 6A
---------------------------------
SUM: 5A 1F 8D D7 99 3F BE 0B 8F97

5080 B7 C8 5E 23 56 23 B9 20 : 52
5088 F5 EB E9 3E FF 21 3E 01 : 66
5090 CD C3 50 D8 C3 2E 50 3A : 33
5098 4B 43 32 4C 43 CD 02 53 : 71
50A0 CD 53 53 CD 6E 50 38 09 : 3F
50A8 E6 01 87 3D CD C3 50 18 : A3
50B0 EC FE 0D CA 2E 50 FE 1B : 58
50B8 20 E3 3A 4C 43 32 4B 43 : 8C
50C0 C3 02 53 21 4B 43 86 37 : 84
50C8 C8 4F 3A 03 50 B9 D8 71 : A6
50D0 C9 21 76 55 09 09 09 09 : D9
50D8 5E 23 56 23 7E 23 66 6F : 70
50E0 22 84 51 CD 42 51 D8 3A : 69
50E8 38 43 28 05 CB 7A 28 01 : 16
50F0 83 D5 CD 74 51 29 11 30 : 54
50F8 43 CD 7F 51 CD 0C 53 D1 : DD
---------------------------------
SUM: 55 EC 08 D8 54 FC 4B 89 93D0

5100 2A 38 43 19 22 38 43 19 : 74
5108 3A 48 43 87 D0 0F 47 7D : EF
5110 D9 CD 74 51 29 CD 1E 20 : 9F
5118 11 28 43 CD FE 51 D9 AF : 20
5120 32 48 43 19 7E FE 0B 20 : 7D
5128 06 32 49 43 06 01 AF B7 : 31
5130 C0 ED 52 77 19 36 0A D9 : A8
5138 11 28 43 CD 7F 51 D9 10 : 02
5140 E2 C9 2A 38 43 19 7E 4F : 36
5148 FE 0A 3F D8 B7 28 18 19 : 2F
5150 7E B7 28 07 D6 0A 37 C0 : 3B
5158 CB F9 FE 71 B7 ED 52 77 : A0
5160 79 CD 6D 51 22 3A 43 79 : 1C
5168 32 48 43 B7 C9 21 00 43 : A1
5170 87 87 6F C9 67 E6 0F 6F : 11
5178 AC 0F 0F 0F 0F 67 C9 CD : E5
---------------------------------
SUM: 5E 32 1B C6 1D CB 58 BC C4A6

5180 82 51 D5 CD 00 00 CD 0E : 50
5188 52 D1 C9 CD 1E 20 2C CD : F0
5190 E6 51 CD CA 51 18 24 2D : 88
5198 CD 1E 20 CD BB 51 CD CA : 7B
51A0 51 18 43 CD 1E 20 24 CD : A8
51A8 F0 51 CD E1 51 18 23 25 : A0
51B0 CD 1E 20 CD D2 51 CD E1 : A9
51B8 51 18 35 3A 48 43 B7 C8 : E2
51C0 D5 ED 5B 3A 43 CD CA 51 : 82
51C8 D1 C9 CD FE 51 3E 1E C3 : D5
51D0 F4 1F 3A 48 43 B7 C8 D5 : 2C
51D8 ED 5B 3A 43 CD E1 51 D1 : 95
51E0 C9 CD FE 51 18 10 CD E2 : BC
51E8 1F 20 1F 1D 20 1E 00 C9 : 82
51F0 CD E2 1F 20 20 00 CD E2 : BD
51F8 1F 1F 1D 1D 00 C9 CD 04 : 12
---------------------------------
SUM: 41 4E E5 54 AF EF 1D B8 E2C4

5200 52 CD F6 51 1A 13 CD F4 : 54
5208 1F 1A 13 C3 F4 1F 3A 4A : A6
5210 43 B7 20 09 01 70 17 0B : B6
5218 78 B1 20 FB C9 0E 0A C3 : E8
5220 FA 53 AF 32 49 43 2A 4B : 2F
5228 43 67 2B 29 29 29 29 29 : A2
5230 11 00 56 19 11 10 40 06 : E7
5238 14 C5 4E 23 06 08 CB 11 : 34
5240 9F E6 0D 12 13 10 F7 C1 : 7F
5248 10 EF EB 06 0C 0E 01 1A : 25
5250 13 B7 28 02 6F 71 0C 10 : F0
5258 F6 E5 11 00 40 21 00 00 : 4D
5260 0E 0C CD 1E 20 24 24 06 : 73
5268 10 1A 13 D9 CD 6D 51 EB : 8C
5270 CD CA 51 D9 10 F3 0D 20 : F1
5278 E9 E1 36 00 22 38 43 CD : 6A
---------------------------------
SUM: 1A 10 5F 99 4E A0 4F 60 6000

5280 02 53 21 00 00 C3 10 53 : 9C
5288 21 04 50 11 00 43 01 04 : CE
5290 00 ED B0 06 09 3E 31 C5 : E0
5298 E5 ED A0 12 13 23 3C ED : E3
52A0 A0 ED A0 E1 C1 10 F0 0E : DD
52A8 04 09 0E 10 ED B0 11 D2 : AB
52B0 55 01 4D 43 1A B7 C8 CD : 4C
52B8 74 51 7D C6 30 02 03 25 : 62
52C0 20 FB 13 18 EF 21 00 40 : 96
52C8 06 C0 36 0D 23 10 FB 21 : 58
52D0 1F 01 11 08 02 3E 3D CD : 83
52D8 3A 54 21 1F 00 11 08 17 : FE
52E0 3E 7B CD 3A 54 11 FB 54 : 74
52E8 EB 5E 23 56 23 EB FE 24 : F2
52F0 CD 1E 20 1A 13 B7 C8 FE : B5
52F8 02 38 ED 28 F2 CD F4 1F : 21
---------------------------------
SUM: EC B8 B1 41 A4 E0 3F B5 974D

5300 18 F1 2A 4B 43 26 00 11 : F8
5308 21 06 18 0A 2A 3C 43 23 : 15
5310 22 3C 43 11 21 09 EB CD : 94
5318 1E 20 EB 0E 20 11 10 27 : 9F
5320 CD 37 53 11 E8 03 CD 37 : 57
5328 53 11 64 00 CD 37 53 1E : 3D
5330 0A CD 37 53 7D 18 0A AF : AF
5338 3C ED 52 30 FB 19 3D 28 : 24
5340 02 0E 30 81 C3 F4 1F 3A : D1
5348 4A 43 B7 20 12 21 25 0B : C7
5350 CD 1E 20 CD 21 20 FE 61 : 78
5358 D8 FE 7B D0 D6 20 C9 2A : 0A
5360 3E 43 7E 23 22 3E 43 C9 : 8E
5368 CD E2 1F 0C 51 75 69 74 : 7D
5370 20 22 46 4C 49 43 4B 22 : CD
5378 0D 00 C3 FA 1F 31 00 43 : 5D
---------------------------------
SUM: 08 09 D8 BB 82 63 A7 C6 017C

5380 3E 28 CD 30 20 CD 88 52 : 2A
5388 CD D0 1F B7 20 FA 3E 0C : D7
5390 CD F4 1F 21 0C 16 CD 1E : 0E
5398 20 11 46 55 CD E5 1F CD : 6A
53A0 18 54 CD 1B 54 3E DB CD : 8E
53A8 1D 54 3E 20 CD 6B 54 21 : 7C
53B0 94 55 CD 7D 54 21 86 55 : 83
53B8 CD 5D 54 CD 9E 54 CD F2 : FC
53C0 53 21 AC 55 CD 7D 54 CD : E0
53C8 9E 54 CD EF 53 3E DB CD : E7
53D0 6B 54 CD 9E 54 CD F8 53 : 96
53D8 CD 1B 54 CD 18 54 3E 20 : D3
53E0 CD 1D 54 CD F5 53 21 4D : C1
53E8 43 22 3E 43 C3 1C 50 CD : E2
53F0 F2 53 CD F5 53 CD F8 53 : 72
53F8 0E 64 CD 01 54 0D 20 FA : BB
---------------------------------
SUM: C7 31 43 97 17 05 22 F2 9260

5400 C9 06 30 CD D0 1F B7 20 : 92
5408 03 10 F8 C9 FE 1B CA 68 : 1F
5410 53 FE 20 CA 1D 50 18 F1 : B1
5418 3E A1 21 3E 6F 11 1B 01 : DA
5420 21 06 0B 06 05 F5 C5 CD : C4
5428 3A 54 0E 02 CD FA 53 C1 : 79
5430 F1 14 14 1C 1C 25 2D 10 : B3
5438 EC C9 32 54 54 3E 2C CD : C6
5440 44 54 3E 2D 4F 43 CD 4C : AE
5448 54 42 CB 99 79 32 58 54 : 51
5450 CD 1E 20 3E 00 CD F4 1F : 29
5458 00 10 F5 C9 23 5E 23 56 : C8
5460 23 7E FE 01 D8 28 F5 ED : 82
5468 A0 18 F6 21 00 40 0E 08 : 25
5470 36 DB 23 06 22 77 23 10 : 06
5478 FC 0D 20 F4 C9 11 25 40 : 5C
---------------------------------
SUM: EF 2E 1D FF 4A 7D AC 3F 721B

5480 0E 06 06 04 C5 4E 23 06 : 5A
5488 08 CB 11 9F E6 5B C6 20 : AA
5490 12 13 10 F5 C1 10 ED 13 : FB
5498 13 13 0D 20 E5 C9 21 40 : 62
54A0 43 06 08 36 02 23 10 FB : B7
54A8 CD B3 54 CD AE 54 3E 2F : 10
54B0 CD B5 54 3E 00 32 CB 54 : 65
54B8 11 C4 55 0E 0E D9 11 00 : 30
54C0 40 0E 08 D9 21 40 43 06 : D9
54C8 08 1A 13 00 87 F5 D9 DC : 66
54D0 E4 54 CD 01 54 21 23 00 : 9E
54D8 19 EB 0C D9 23 F1 10 EC : F9
54E0 0D 20 DA C9 D9 7E 34 D9 : 34
54E8 61 6F CD 1E 20 26 00 2D : 2E
54F0 2D 19 7E CD F4 1F 3E DB : BD
54F8 C3 F4 1F 21 02 46 4C 49 : D4
---------------------------------
SUM: CC 2C 71 8F 1D 54 2E EF CC6E

5500 43 4B 01 21 05 52 4F 55 : AB
5508 4E 44 01 22 08 53 54 45 : A9
5510 50 01 21 0B 4B 45 59 3A : A0
5518 02 02 52 29 65 74 72 79 : 43
5520 02 02 4E 29 65 78 74 02 : CE
5528 02 42 29 61 63 6B 02 02 : A0
5530 53 29 6B 69 70 02 02 51 : 15
5538 29 75 69 74 00 04 17 43 : D9
5540 4C 45 41 52 2E 20 48 69 : 23
5548 74 20 5B 53 50 41 43 45 : 5B
5550 5D 20 6B 65 79 21 00 32 : 19
5558 34 36 38 4B 4A 4C 49 1F : EB
5560 1D 1C 1E 52 2E 50 4E 8E : 03
5568 50 42 8B 50 53 97 50 51 : F8
5570 68 53 00 00 00 00 F0 FF : AA
5578 AF 51 01 00 8B 51 FF FF : DB
---------------------------------
SUM: 38 31 A9 D5 42 4D 5E C1 3EB2

5580 97 51 10 00 A3 51 25 40 : 51
5588 66 6F 72 01 E0 40 53 57 : 12
5590 4F 52 44 00 07 E0 3F 3F : 4A
5598 07 80 27 3C 07 E7 27 3F : 3E
55A0 07 E7 3F 3F 01 E0 3F 0F : 9B
55A8 07 E0 3F 3F 7C 83 8E 22 : 14
55B0 40 81 11 24 78 81 10 28 : 27
55B8 40 81 10 34 40 81 11 22 : F9
55C0 40 F3 8E 22 F0 C8 F4 AA : 39
55C8 F0 CC F0 0F 33 0F 55 0B : 5D
55D0 37 0F 28 46 52 26 18 24 : 68
55D8 18 14 12 14 12 26 12 46 : E2
55E0 28 44 18 14 12 16 12 54 : 26
55E8 12 14 18 14 18 56 32 46 : 38
55F0 18 24 12 14 18 76 18 54 : 5C
55F8 12 14 68 52 66 68 64 00 : 12
---------------------------------
SUM: C4 CD EE 2C F5 2A FF 9D A292

5600 80 01 80 01 BE 7D A0 45 : 22
5608 A6 45 A2 45 BE 7D 80 01 : 8E
5610 80 01 80 01 00 00 00 8B : 8D
5618 8D 89 00 00 00 83 22 64 : 1F
5620 80 01 80 01 86 61 84 01 : 6E
5628 84 21 84 21 87 E1 80 01 : 33
5630 80 01 80 01 00 00 00 82 : 84
5638 72 62 52 42 32 34 66 84 : B8
5640 84 21 84 21 84 21 84 21 : 94
5648 84 01 84 21 80 21 84 21 : 70
5650 84 21 84 21 00 00 00 00 : 4A
5658 8E 8D 8C 7C 6C 34 AE AC : 1D
5660 80 01 9F F1 80 05 AF E5 : 2A
5668 E0 07 80 01 AF F5 80 01 : 8D
5670 8F F1 80 01 67 63 84 58 : A7
5678 7C 00 00 00 32 57 68 6A : D7
---------------------------------
SUM: AE 1E 2F 7E F3 1D 7D D3 EAEB

5680 8F F1 81 81 85 A1 80 01 : 29
5688 F3 CF F3 CF 80 01 85 A1 : 2B
5690 81 81 8F F1 2D 3D 4D 00 : 39
5698 00 00 82 00 92 44 AE AD : B3
56A0 CC 33 80 01 80 01 8F B1 : 41
56A8 88 01 80 11 8D F1 80 01 : 19
56B0 80 01 CC 33 64 00 00 49 : 2D
56B8 00 76 5B 5C 5D 67 58 72 : BB
56C0 80 01 8D 81 90 61 90 19 : 29
56C8 A0 05 A0 05 98 09 86 09 : 7A
56D0 81 B1 80 01 00 00 00 00 : B3
56D8 66 67 68 69 6A 47 A1 46 : 36
56E0 87 E1 80 09 94 29 80 01 : 2F
56E8 F1 8F F1 8F 80 01 94 29 : 3E
56F0 80 01 83 C1 89 39 3B 00 : C2
56F8 00 36 00 00 86 43 AE 4B : F8
---------------------------------
SUM: D6 B1 B5 2B 47 D3 1B 99 9436

5700 A0 01 A0 71 BC 01 82 85 : 76
5708 80 45 80 05 80 85 BC 01 : 0C
5710 A0 71 A0 01 00 00 27 57 : 30
5718 25 97 67 95 71 38 11 5A : CC
5720 80 07 A1 07 A1 01 BB 79 : 05
5728 80 03 83 C1 B8 1D 80 01 : 1D
5730 9C 39 83 C1 00 2B 25 24 : 8D
5738 00 00 87 88 89 3A AE 86 : 06
5740 A1 85 A1 85 80 01 8E 11 : 6C
5748 80 11 80 11 8E F1 A8 11 : 5A
5750 AA 91 88 11 00 00 4A 00 : 1E
5758 9D 8D 6D 5D 4D 8C AA 49 : C0
5760 8F 01 89 1D 89 01 83 71 : B4
5768 89 41 88 15 88 85 80 85 : 79
5770 88 95 88 81 22 32 42 52 : 0E
5778 62 72 82 92 00 8B 11 86 : 0A
---------------------------------
SUM: EB 8E 86 66 1D 02 04 94 22A5

5780 88 11 A8 15 A0 01 A9 91 : 31
5788 89 91 89 91 89 95 80 05 : D7
5790 A8 15 88 11 00 00 00 27 : 7D
5798 28 00 97 98 99 63 6C 96 : 55
57A0 A0 01 83 C1 A3 C1 80 41 : 0A
57A8 A2 01 83 C1 BB DD A0 05 : 24
57B0 A0 05 B9 9D 00 00 2B 2C : 52
57B8 2D 4B 4C 4D 00 5C A1 3C : 4A
57C0 80 01 BB D5 A0 01 A3 DF : 34
57C8 A3 01 A0 01 A3 C1 A2 41 : 8C
57D0 BA 01 80 01 8A 44 00 25 : 2F
57D8 69 8C 3B 3A 6B 6A 68 8D : 34
57E0 81 E3 A9 F9 81 FD A9 65 : 92
57E8 89 25 C0 01 C0 29 8F A9 : 90
57F0 80 01 E2 39 96 00 32 34 : 98
57F8 00 73 75 77 79 83 97 36 : 28
---------------------------------
SUM: C0 14 31 76 A8 0C 2F 4B 8DB5
```

リスト2　FLICKソースリスト(参考)

```
                              1 ;//////////////////////////////////////
                              2 ;
                              3 ;        F L I C K   ver 1.1
                              4 ;
                              5 ;         for  S-OS "SWORD"
                              6 ;
                              7 ;//////////////////////////////////////
                              8
                              9         OFFSET  $C000-$5000
                             10         ORG     $5000
                             11
(1FFA)                       12 #HOT    EQU     1FFAH
(1FF4)                       13 #PRINT  EQU     1FF4H
(1FE5)                       14 #MSX    EQU     1FE5H
(1FE2)                       15 #MPRNT  EQU     1FE2H
(1FD0)                       16 #GETKY  EQU     1FD0H
(1FCA)                       17 #INKEY  EQU     1FCAH
(201E)                       18 #LOC    EQU     201EH
(2021)                       19 #FLGET  EQU     2021H
(2030)                       20 #WIDCH  EQU     2030H
                             21
(4000)                       22 BUFFER  EQU     4000H
(4300)                       23 STACK   EQU     4300H
(4300)                       24 CHRBUF  EQU     4300H
(4338)                       25 POS     EQU     CHRBUF+14*4
(433A)                       26 MOVADR  EQU     POS    +2
(433C)                       27 STEP    EQU     MOVADR+2
(433E)                       28 DKEYAD  EQU     STEP   +2
(4340)                       29 DPOS    EQU     DKEYAD+2
(4348)                       30 MOVFLG  EQU     DPOS   +8
(4349)                       31 CLRFLG  EQU     MOVFLG+1
(434A)                       32 DEMOF   EQU     CLRFLG+1
(434B)                       33 STAGE   EQU     DEMOF  +1
(434C)                       34 STBAK   EQU     STAGE  +1
(434D)                       35 KEYBUF  EQU     STBAK  +1
                             36
                             37         ;//////////////////////////////
                             38
 5000  C3 537D               39         JP DEMO
                             40 MAX:
 5003  10                    41         DB 16
                             42 CHRDAT:
 5004  20 20 20 20           43         DB '    '
 5008  A2 20 4C A3           44         DB '｢ L｣'
 500C  7B 7B 55 55           45         DB '██UU'
 5010  A5 A5 A5 A5           46         DB '････'
 5014  3C 3E 22 22           47         DB '<>""'
 5018  DB 7B 7B 7B           48         DB '□███'
                             49         ;//////////////////////////////
                             50 START1:
 501C  3E                    51         DB [LD A]
                             52 START2:
 501D  AF                    53         XOR A
 501E  32 434A               54         LD (DEMOF),A
 5021  3E 0C CD 1FF4         55         LD A,0CH:CALL #PRINT
 5026  CD 52C5               56         CALL INIT2
 5029  3E 01 32 434B         57         LD A,1:LD (STAGE),A
                             58 RETRY:
 502E  31 4300               59         LD SP,STACK
 5031  CD 5222               60         CALL SCREEN
                             61         ;---------------------
                             62 MAIN1:
 5034  CD 5347               63         CALL KEY1
 5037  CD 506E 30 05         64           CALL MOVKEY?:JR NC,MAIN2
 503C  CD 507D               65           CALL FUNC1
 503F  18 F3                 66         JR MAIN1
                             67 MAIN2:
 5041  CD 50D1               68           CALL MOVE
 5044  3A 4349               69           LD A,(CLRFLG)
 5047  B7 28 EA              70         IF A=0 JR MAIN1
                             71         ;---------------------
                             72 CLEAR:
 504A  11 553D               73         LD DE,CLRMSG
 504D  CD 52E8               74         CALL PRTMSG
 5050  3A 434A B7 20 12      75         LD A,(DEMOF):IF A<>0 JR CLEAR2
                             76 CLEAR1:
 5056  CD 1FCA               77         CALL #INKEY
 5059  FE 1B CA 5368         78         IF A=1BH  JP QUIT
 505E  FE 20 20 F4           79         IF A<>20H JR CLEAR1
 5062  CD 508E               80         CALL NEXT
 5065  C3 5388               81         JP DEMO1
                             82 CLEAR2:
 5068  CD 53EF               83         CALL DWAIT_8
 506B  C3 538E               84         JP DEMO2
                             85         ;---------------------
                             86 MOVKEY?:
 506E  21 5557 01 000C       87         LD HL,MOVKEY:LD BC,12
 5074  ED B1 37 C0           88         CPIR:SCF:RET NZ
 5078  79 E6 03 4F           89         LD A,C:AND 03H:LD C,A
 507C  C9                    90         RET
                             91         ;---------------------
                             92 FUNC1:
 507D  4F                    93         LD C,A
                             94 FUNC2:
 507E  7E 23 B7 C8           95         LD A,(HL):INC HL:IF A=0 RET
 5082  5E 23                 96         LD E,(HL):INC HL
 5084  56 23                 97         LD D,(HL):INC HL
 5086  B9 20 F5              98         IF A<>C JR FUNC2
 5089  EB                    99         EX DE,HL
 508A  E9                   100         JP (HL)
                            101         ;//////////////////////////////
                            102 BACK:
 508B  3E FF                103         LD A,-1
 508D  21                   104          DB [LD HL]
                            105 NEXT:
 508E  3E 01                106         LD A,1
 5090  CD 50C3 D8           107         CALL CHGSTG:RET C
 5094  C3 502E              108         JP RETRY
                            109         ;---------------------
                            110 SKIP:
 5097  3A 434B 32 434C      111         LD A,(STAGE):LD (STBAK),A
                            112 SKP1:
 509D  CD 5302 CD 5353      113         CALL PRTSTG :CALL KEY2
 50A3  CD 506E 38 09        114         CALL MOVKEY?:JR C,SKP2
 50A8  E6 01 87 3D          115           AND 01H:ADD A,A:DEC A
 50AC  CD 50C3              116           CALL CHGSTG
 50AF  18 EC                117         JR SKP1
                            118 SKP2:
 50B1  FE 0D CA 502E        119           IF A=0DH  JP RETRY
 50B6  FE 1B 20 E3          120           IF A<>1BH JR SKP1
 50BA  3A 434C 32 434B      121         LD A,(STBAK):LD (STAGE),A
 50C0  C3 5302              122         JP PRTSTG
                            123         ;---------------------
                            124 CHGSTG:
 50C3  21 434B              125         LD HL,STAGE
 50C6  86 37 C8             126         ADD A,(HL):SCF:RET Z
 50C9  4F 3A 5003           127         LD C,A:LD A,(MAX)
 50CD  B9 D8                128         CP C:RET C
 50CF  71                   129         LD (HL),C
 50D0  C9                   130         RET
                            131         ;//////////////////////////////
                            132 MOVE:
 50D1  21 5576              133         LD HL,MOVDAT
 50D4  09 09                134         ADD HL,BC:ADD HL,BC
 50D6  09 09                135         ADD HL,BC:ADD HL,BC
 50D8  5E 23                136         LD E,(HL):INC HL
 50DA  56 23                137         LD D,(HL):INC HL
 50DC  7E 23                138         LD A,(HL):INC HL
 50DE  66 6F 22 5184        139         LD H,(HL):LD L,A:LD (MOVCHR1+2),HL
 50E3  CD 5142 D8           140         CALL CHECK:RET C
 50E7  3A 4338 28 05        141         LD A,(POS):JR Z,MOVE1
 50EC  CB 7A 28 01 83       142           IF [BIT 7,D]<>0 THEN ADD A,E
                            143 MOVE1:
 50F1  D5                   144         PUSH DE
 50F2  CD 5174 29           145           CALL CALHALF:ADD HL,HL
 50F6  11 4330              146           LD DE,CHRBUF+12*4
 50F9  CD 517F              147           CALL MOVCHR
 50FC  CD 530C              148           CALL INCSTEP
 50FF  D1                   149         POP DE
 5100  2A 4338 19           150         LD HL,(POS):ADD HL,DE
 5104  22 4338 19           151         LD (POS),HL:ADD HL,DE
 5108  3A 4348 87 D0        152         LD A,(MOVFLG):ADD A,A:RET NC
 510D  0F 47 7D             153         RRCA:LD B,A:LD A,L
 5110  D9                   154         EXX
 5111  CD 5174 29           155           CALL CALHALF:ADD HL,HL
 5115  CD 201E              156           CALL #LOC
 5118  11 4328              157           LD DE,CHRBUF+10*4
 511B  CD 51FE              158           CALL PUTCHR
 511E  D9                   159         EXX
 511F  AF 32 4348           160         XOR A:LD (MOVFLG),A
                            161 MOVE2:
 5123  19 7E                162         ADD HL,DE:LD A,(HL)
 5125  FE 0B 20 06 32       163         IF A=11 THEN LD (CLRFLG),A:LD B,1
 512A  4349 06 01 AF                    :XOR A
 512F  B7 C0                164         IF A<>0 RET
 5131  ED 52 77             165         SBC HL,DE:LD (HL),A
 5134  19 36 0A             166         ADD HL,DE:LD (HL),10
 5137  D9                   167         EXX
 5138  11 4328              168           LD DE,CHRBUF+10*4
 513B  CD 517F              169           CALL MOVCHR
 513E  D9                   170         EXX
 513F  10 E2                171         DJNZ MOVE2
 5141  C9                   172         RET
                            173         ;---------------------
                            174 CHECK:
 5142  2A 4338              175         LD HL,(POS)
 5145  19 7E 4F             176         ADD HL,DE:LD A,(HL):LD C,A
 5148  FE 0A 3F D8          177         CP 10:CCF:RET C
 514C  B7 28 18             178         IF A=0 JR CHK3
 514F  19 7E                179         ADD HL,DE:LD A,(HL)
 5151  B7 28 07             180         IF A=0 JR CHK2
 5154  D6 0A 37 C0          181           SUB 10:SCF:RET NZ
 5158  CB F9 FE             182           SET 7,C:DB [CP]
                            183 CHK2:
 515B  71 B7 ED 52          184         LD (HL),C:SUB HL,DE
 515F  77                   185         LD (HL),A
 5160  79 CD 516D           186         LD A,C:CALL CALCHR
 5164  22 433A              187         LD (MOVADR),HL
                            188 CHK3:
 5167  79 32 4348           189         LD A,C:LD (MOVFLG),A
 516B  B7                   190         OR A
 516C  C9                   191         RET
                            192         ;---------------------
                            193 CALCHR:
 516D  21 4300              194         LD HL,CHRBUF
 5170  87 87                195         ADD A,A:ADD A,A
 5172  6F                   196         LD L,A
 5173  C9                   197         RET
                            198 CALHALF:
 5174  67 E6 0F 6F          199         LD H,A:AND 0FH:LD L,A
 5178  AC 0F 0F 0F 0F       200         XOR H :RRCA:RRCA:RRCA:RRCA
 517D  67                   201         LD H,A
 517E  C9                   202         RET
                            203         ;---------------------
                            204 MOVCHR:
 517F  CD 5182              205         CALL MOVCHR1
                            206 MOVCHR1:
 5182  D5                   207         PUSH DE
 5183  CD 0000 CD 520E      208           CALL 0000:CALL WAIT
 5189  D1                   209         POP DE
 518A  C9                   210         RET
                            211 RGTCHR:
 518B  CD 201E 2C           212         CALL #LOC:INC L
 518F  CD 51E6              213         CALL PUTSP_H
 5192  CD 51CA              214         CALL CPUT2_H
 5195  18 24                215         JR   CPUT1_H
                            216 LFTCHR:
 5197  2D CD 201E           217         DEC L:CALL #LOC
 519B  CD 51BB              218         CALL CPUT1_H
 519E  CD 51CA              219         CALL CPUT2_H
 51A1  18 43                220         JR   PUTSP_H
                            221 DWNCHR:
 51A3  CD 201E 24           222         CALL #LOC:INC H
 51A7  CD 51F0              223         CALL PUTSP_V
 51AA  CD 51E1              224         CALL CPUT2_V
 51AD  18 23                225         JR   CPUT1_V
                            226 UPCHR:
 51AF  25 CD 201E           227         DEC H:CALL #LOC
 51B3  CD 51D2              228         CALL CPUT1_V
 51B6  CD 51E1              229         CALL CPUT2_V
 51B9  18 35                230         JR   PUTSP_V
                            231         ;---------------------
                            232 CPUT1_H:
 51BB  3A 4348 B7 C8        233         LD A,(MOVFLG):IF A=0 RET
 51C0  D5                   234         PUSH DE
 51C1  ED 5B 433A           235           LD DE,(MOVADR)
 51C5  CD 51CA              236           CALL CPUT2_H
 51C8  D1                   237         POP DE
 51C9  C9                   238         RET
                            239 CPUT2_H:
 51CA  CD 51FE              240         CALL PUTCHR
 51CD  3E 1E                241         LD A,1EH
 51CF  C3 1FF4              242         JP #PRINT
                            243         ;---------------------
                            244 CPUT1_V:
 51D2  3A 4348 B7 C8        245         LD A,(MOVFLG):IF A=0 RET
 51D7  D5                   246         PUSH DE
 51D8  ED 5B 433A           247           LD DE,(MOVADR)
 51DC  CD 51E1              248           CALL CPUT2_V
 51DF  D1                   249         POP DE
 51E0  C9                   250         RET
                            251 CPUT2_V:
 51E1  CD 51FE              252         CALL PUTCHR
 51E4  18 10                253         JR LF
                            254         ;---------------------
                            255 PUTSP_H:
 51E6  CD 1FE2              256         CALL #MPRNT
 51E9  20 1F 1D             257          DB 20H,1FH,1DH
 51EC  20 1E 00             258          DB 20H,1EH,0
 51EF  C9                   259         RET
                            260 PUTSP_V:
 51F0  CD 1FE2              261         CALL #MPRNT
 51F3  20 20 00             262          DB 20H,20H,0
                            263 LF:
 51F6  CD 1FE2              264         CALL #MPRNT
 51F9  1F 1D 1D 00          265          DB 1FH,1DH,1DH,0
 51FD  C9                   266         RET
                            267         ;---------------------
                            268 PUTCHR:
 51FE  CD 5204 CD 51F6      269         CALL PUT1:CALL LF
                            270 PUT1:
 5204  1A 13                271         LD A,(DE):INC DE
 5206  CD 1FF4              272         CALL #PRINT
 5209  1A 13                273         LD A,(DE):INC DE
 520B  C3 1FF4              274         JP #PRINT
                            275         ;//////////////////////////////
                            276 WAIT:
 520E  3A 434A B7 20 09     277         LD A,(DEMOF):IF A<>0 JR WAIT2
 5214  01 1770              278         LD BC,6000
                            279 WAIT1:
 5217  0B                   280         DEC BC
 5218  78 B1 20 FB          281         IF BC<>0 JR WAIT1
 521C  C9                   282         RET
                            283 WAIT2:
 521D  0E 0A                284         LD C,10
 521F  C3 53FA              285         JP DWAIT_C
                            286         ;//////////////////////////////
                            287 SCREEN:
 5222  AF 32 4349           288         XOR A:LD (CLRFLG),A
 5226  2A 434B 67           289         LD HL,(STAGE):LD H,A
 522A  2B 29                290         DEC HL   :ADD HL,HL
 522C  29 29                291         ADD HL,HL:ADD HL,HL
 522E  29 29                292         ADD HL,HL:ADD HL,HL
 5230  11 5600 19           293         LD DE,STGDAT:ADD HL,DE
 5234  11 4010              294         LD DE,BUFFER+16
 5237  06 14                295         LD B,2*10
                            296 SCRN1:
 5239  C5                   297         PUSH BC
 523A  4E 23 06 08          298           LD C,(HL):INC HL:LD B,8
                            299 SCRN2:
```


▶年がいもなく，荻野目ちゃんのコンサートに行ってきました。彼女があまりにも普通の女の子だったのであっけとられてしまいましたが，荻野目ちゃんの目がとても印象的で，歌もうまかった。タダで最前列のほぼ中央に座れるとは，なんて幸せ者なんだろう。
藤崎　光輝　(20)　静岡県

```
523E  CB 11 9F E6 0D      300             RL C:SBC A,A:AND 0DH
5243  12 13               301             LD (DE),A:INC DE
5245  10 F7               302             DJNZ SCRN2
5247  C1                  303           POP BC
5248  10 EF               304           DJNZ SCRN1
524A  EB                  305           EX DE,HL
524B  06 0C 0E 01         306           LD B,12:LD C,1
                          307 SCRN3:
524F  1A 13               308           LD A,(DE):INC DE
5251  B7 28 02 6F 71      309           IF A<>0 THEN LD L,A:LD (HL),C
5256  0C                  310           INC C
5257  10 F6               311           DJNZ SCRN3
                          312
5259  E5                  313           PUSH HL
525A  11 4000 21 0000     314             LD DE,BUFFER:LD HL,0
5260  0E 0C               315             LD C,12
                          316 SCRN4:
5262  CD 201E 24 24       317             CALL #LOC:INC H,H
5267  06 10               318             LD B,16
                          319 SCRN5:
5269  1A 13               320             LD A,(DE):INC DE
526B  D9                  321             EXX
526C  CD 516D EB          322               CALL CALCHR:EX DE,HL
5270  CD 51CA             323               CALL CPUT2_H
5273  D9                  324             EXX
5274  10 F3               325             DJNZ SCRN5
5276  0D 20 E9            326             DEC C:JR NZ,SCRN4
5279  E1                  327           POP HL
527A  36 00 22 4338       328           LD (HL),0:LD (POS),HL
527F  CD 5302             329           CALL PRTSTG
5282  21 0000             330           LD HL,0
5285  C3 5310             331           JP SETSTEP
                          332           ;///////////////////////////////
                          333 INIT1:
5288  21 5004             334           LD HL,CHRDAT
528B  11 4300             335           LD DE,CHRBUF
528E  01 0004 ED B0       336           LD BC,4:LDIR
5293  06 09 3E 31         337           LD B,9:LD A,'1'
                          338 MAKE1:
5297  C5 E5               339           PUSH BC,HL
5299  ED A0 12            340             LDI:LD (DE),A
529C  13 23 3C            341             INC DE,HL,A
529F  ED A0 ED A0         342             LDI:LDI
52A3  E1 C1               343           POP HL,BC
52A5  10 F0               344           DJNZ MAKE1
52A7  0E 04 09            345           LD C,4:ADD HL,BC
52AA  0E 10 ED B0         346           LD C,4*4:LDIR
                          347
52AE  11 55D2             348           LD DE,KEYDAT
52B1  01 434D             349           LD BC,KEYBUF
                          350 MAKE4:
52B4  1A B7 C8            351           LD A,(DE):IF A=0 RET
52B7  CD 5174             352           CALL CALHALF
52BA  7D C6 30            353           LD A,L:ADD A,'0'
                          354 MAKE5:
52BD  02 03               355           LD (BC),A:INC BC
52BF  25 20 FB            356           DEC H:JR NZ,MAKE5
52C2  13                  357           INC DE
52C3  18 EF               358           JR MAKE4
                          359           ;---------------------
                          360 INIT2:
52C5  21 4000 06 C0       361           LD HL,BUFFER:LD B,16*12
                          362 INIT21:
52CA  36 0D 23            363           LD (HL),13:INC HL
52CD  10 FB               364           DJNZ INIT21
52CF  21 011F 11 0208     365           LD HL,011FH:LD DE,0208H
52D5  3E 3D CD 543A       366           LD A,'=':CALL PRTBOX4
52DA  21 001F 11 1708     367           LD HL,001FH:LD DE,1708H
52E0  3E 7B CD 543A       368           LD A,7BH:CALL PRTBOX4
52E5  11 54FB             369           LD DE,RCMSG
                          370           ;---------------------
                          371 PRTMSG:
52E8  EB                  372           EX DE,HL
52E9  5E 23               373           LD E,(HL):INC HL
52EB  56 23               374           LD D,(HL):INC HL
52ED  EB                  375           EX DE,HL
52EE  FE                  376           DB [CP]
                          377 PRTM1:
52EF  24                  378           INC H
52F0  CD 201E             379           CALL #LOC
                          380 PRTM2:
52F3  1A 13               381           LD A,(DE):INC DE
52F5  B7 C8               382           IF A=0 RET
52F7  FE 02 38 ED         383           CP 2:JR C,PRTMSG
52FB  28 F2               384                JR Z,PRTM1
52FD  CD 1FF4             385           CALL #PRINT
5300  18 F1               386           JR PRTM2
                          387           ;---------------------
                          388 PRTSTG:
5302  2A 434B 26 00       389           LD HL,(STAGE):LD H,0
5307  11 0621             390           LD DE,0621H
530A  18 0A               391           JR PRTNUM
                          392           ;---------------------
                          393 INCSTEP:
530C  2A 433C 23          394           LD HL,(STEP):INC HL
                          395 SETSTEP:
5310  22 433C             396           LD (STEP),HL
5313  11 0921             397           LD DE,0921H
                          398           ;---------------------
                          399 PRTNUM:
5316  EB CD 201E EB       400           EX DE,HL:CALL #LOC:EX DE,HL
531B  0E 20               401           LD C,20H
531D  11 2710 CD 5337     402           LD DE,10000:CALL PRTD1
5323  11 03E8 CD 5337     403           LD DE,1000 :CALL PRTD1
5329  11 0064 CD 5337     404           LD DE,100  :CALL PRTD1
532F  1E 0A CD 5337       405           LD E,10    :CALL PRTD1
5334  7D                  406           LD A,L
5335  18 0A               407           JR PRTD3
                          408 PRTD1:
5337  AF                  409           XOR A
5338  3C ED 52            410 PRTD2:  INC A:SBC HL,DE
533B  30 FB               411           JR NC,PRTD2
533D  19                  412           ADD HL,DE
533E  3D 28 02            413           DEC A:JR Z,PRTD4
                          414 PRTD3:
5341  0E 30               415           LD C,'0'
                          416 PRTD4:
5343  81                  417           ADD A,C
5344  C3 1FF4             418           JP #PRINT
                          419           ;---------------------
                          420 KEY1:
5347  3A 434A B7 20 12    421           LD A,(DEMOF):IF A<>0 JR DKEY
534D  21 0B25 CD 201E     422           LD HL,0B25H:CALL #LOC
                          423 KEY2:
5353  CD 2021             424           CALL #FLGET
5356  FE 61 D8            425           CP 'a'  :RET C
5359  FE 7B D0            426           CP 'z'+1:RET NC
535C  D6 20               427           SUB 20H
535E  C9                  428           RET
                          429 DKEY:
535F  2A 433E             430           LD HL,(DKEYAD)
5362  7E 23               431           LD A,(HL):INC HL
5364  22 433E             432           LD (DKEYAD),HL
5367  C9                  433           RET
                          434           ;///////////////////////////////
                          435 QUIT:
5368  CD 1FE2             436           CALL #MPRNT
536B  0C 51 75 69 74 20   437            DB 0CH,'Quit "FLICK"',0DH,0
5371  22 46 4C 49 43 4B
5377  22 0D 00
537A  C3 1FFA             438           JP #HOT
                          439
                          440 ;=====================================
                          441
                          442 DEMO:
537D  31 4300             443           LD SP,STACK
5380  3E 28 CD 2030       444           LD A,40:CALL #WIDCH
5385  CD 5288             445           CALL INIT1
                          446 DEMO1:
5388  CD 1FD0 B7 20 FA    447           CALL #GETKY:IF A<>0 JR DEMO1
                          448 DEMO2:
538E  3E 0C CD 1FF4       449           LD A,0CH     :CALL #PRINT
5393  21 160C CD 201E     450           LD HL,160CH  :CALL #LOC
5399  11 5546 CD 1FE5     451           LD DE,HITMSG :CALL #MSX
539F  CD 5418 CD 541B     452           CALL PRTBOX11:CALL PRTBOX12
53A5  3E DB CD 541D       453           LD A,'□'     :CALL PRTBOX2
53AA  3E 20 CD 546B       454           LD A,' '     :CALL SETDT1
53AF  21 5594 CD 547D     455           LD HL,TTLDAT1:CALL SETDT2
53B5  21 5586 CD 545D     456           LD HL,MSGDAT :CALL SETSTR
53BB  CD 549E CD 53F2     457           CALL WAVESUB :CALL DWAIT_4
53C1  21 55AC CD 547D     458           LD HL,TTLDAT2:CALL SETDT2
53C7  CD 549E CD 53EF     459           CALL WAVESUB :CALL DWAIT_8
53CD  3E DB CD 546B       460           LD A,'□'     :CALL SETDT1
53D2  CD 549E CD 53F8     461           CALL WAVESUB :CALL DWAIT_1
53D8  CD 541B CD 5418     462           CALL PRTBOX12:CALL PRTBOX11
53DE  3E 20 CD 541D       463           LD A,' '     :CALL PRTBOX2
53E3  CD 53F5             464           CALL DWAIT_2
53E6  21 434D 22 433E     465           LD HL,KEYBUF :LD (DKEYAD),HL
53EC  C3 501C             466           JP START1
                          467           ;---------------------
                          468 DWAIT_8:
53EF  CD 53F2             469           CALL DWAIT_4
                          470 DWAIT_4:
53F2  CD 53F5             471           CALL DWAIT_2
                          472 DWAIT_2:
53F5  CD 53F8             473           CALL DWAIT_1
                          474 DWAIT_1:
53F8  0E 64               475           LD C,100
                          476 DWAIT_C:
53FA  CD 5401             477           CALL DWAIT
53FD  0D 20 FA            478           DEC C:JR NZ,DWAIT_C
5400  C9                  479           RET
                          480           ;---------------------
                          481 DWAIT:
5401  06 30               482           LD B,30H
                          483 DWAIT1:
5403  CD 1FD0             484           CALL #GETKY
5406  B7 20 03            485           IF A<>0 JR DWAIT3
5409  10 F8               486 DWAIT2: DJNZ DWAIT1
540B  C9                  487           RET
                          488 DWAIT3:
540C  FE 1B CA 5368       489             IF A=1BH JP QUIT
5411  FE 20 CA 501D       490             IF A=20H JP START2
5416  18 F1               491           JR DWAIT2
                          492           ;---------------------
                          493 PRTBOX11:
5418  3E A1               494           LD A,'.'
541A  21                  495            DB [LD HL]
                          496 PRTBOX12:
541B  3E 6F               497           LD A,'o'
                          498 PRTBOX2:
541D  11 011B             499           LD DE,011BH
5420  21 0B06 06 05       500           LD HL,0B06H:LD B,5
                          501 PRTBOX3:
5425  F5 C5               502           PUSH AF,BC
5427  CD 543A             503             CALL PRTBOX4
542A  0E 02 CD 53FA       504             LD C,2:CALL DWAIT_C
542F  C1 F1               505           POP BC,AF
5431  14 14 1C 1C 25 2D   506           INC D,D,E,E:DEC H,L
5437  10 EC               507           DJNZ PRTBOX3
5439  C9                  508           RET
                          509 PRTBOX4:
543A  32 5454             510           LD (PRTBOX8+1),A
543D  3E 2C CD 5444       511           LD A,[INC L]:CALL PRTBOX5
5442  3E 2D               512           LD A,[DEC L]
                          513 PRTBOX5:
5444  4F                  514           LD C,A
5445  43 CD 544C          515           LD B,E:CALL PRTBOX6
5449  42 CB 99            516           LD B,D:RES 3,C
                          517 PRTBOX6:
544C  79 32 5458          518           LD A,C:LD (PRTBOX9),A
                          519 PRTBOX7:
5450  CD 201E             520           CALL #LOC
                          521 PRTBOX8:
5453  3E 00 CD 1FF4       522           LD A,00:CALL #PRINT
                          523 PRTBOX9:
5458  00                  524           NOP
5459  10 F5               525           DJNZ PRTBOX7
545B  C9                  526           RET
                          527           ;---------------------
545C  23                  528           INC HL
                          529 SETSTR:
545D  5E 23               530           LD E,(HL):INC HL
545F  56 23               531           LD D,(HL):INC HL
                          532 SETSTR1:
5461  7E                  533           LD A,(HL)
5462  FE 01 D8 28 F5      534           CP 1:RET C:JR Z,SETSTR-1
5467  ED A0               535           LDI
5469  18 F6               536           JR SETSTR1
                          537           ;---------------------
                          538 SETDT1:
546B  21 4000 0E 08       539           LD HL,BUFFER:LD C,8
                          540 SETD11:
5470  36 DB 23            541           LD (HL),'□':INC HL
5473  06 22               542           LD B,34
                          543 SETD12:
5475  77 23               544           LD (HL),A:INC HL
5477  10 FC               545           DJNZ SETD12
5479  0D 20 F4            546           DEC C:JR NZ,SETD11
547C  C9                  547           RET
                          548           ;---------------------
                          549 SETDT2:
547D  11 4025 0E 06       550           LD DE,BUFFER+35+2:LD C,6
                          551 SETD21:
5482  06 04               552           LD B,4
                          553 SETD22:
5484  C5                  554           PUSH BC
5485  4E 23               555             LD C,(HL):INC HL
5487  06 08               556             LD B,8
                          557 SETD23:
5489  CB 11 9F            558             RL C:SBC A,A
548C  E6 5B C6 20         559             AND 7BH-20H:ADD A,20H
5490  12 13               560             LD (DE),A:INC DE
5492  10 F5               561             DJNZ SETD23
5494  C1                  562           POP BC
5495  10 ED               563           DJNZ SETD22
5497  13 13 13            564           INC DE,DE,DE
549A  0D 20 E5            565           DEC C:JR NZ,SETD21
549D  C9                  566           RET
                          567           ;---------------------
                          568 WAVESUB:
549E  21 4340 06 08       569           LD HL,DPOS:LD B,8
                          570 WAVE0:
54A3  36 02 23            571           LD (HL),2:INC HL
54A6  10 FB               572           DJNZ WAVE0
54A8  CD 54B3 CD 54AE     573           CALL WAVE2 :CALL WAVE1
                          574 WAVE1:
54AE  3E 2F CD 54B5       575           LD A,[CPL]:CALL WAVE3
                          576 WAVE2:
54B3  3E 00               577           LD A,[NOP]
                          578 WAVE3:
54B5  32 54CB             579           LD (WAVE5),A
54B8  11 55C4 0E 0E       580           LD DE,WAVDAT:LD C,7*2
                          581 WAVE4:
54BD  D9                  582           EXX
54BE  11 4000 0E 08       583           LD DE,BUFFER:LD C,8
54C3  D9                  584           EXX
54C4  21 4340 06 08       585           LD HL,DPOS:LD B,8
54C9  1A 13               586           LD A,(DE):INC DE
                          587 WAVE5:
54CB  00                  588           NOP
                          589 WAVE6:
54CC  87 F5               590           ADD A,A:PUSH AF
54CE  D9                  591             EXX
54CF  DC 54E4             592             CALL C,WAVE7
54D2  CD 5401             593             CALL DWAIT
54D5  21 0023 19          594             LD HL,35:ADD HL,DE
54D9  EB 0C               595             EX DE,HL:INC C
54DB  D9 23               596             EXX:INC HL
54DD  F1                  597           POP AF
54DE  10 EC               598           DJNZ WAVE6
54E0  0D 20 DA            599           DEC C:JR NZ,WAVE4
54E3  C9                  600           RET
                          601 WAVE7:
54E4  D9                  602           EXX
54E5  7E 34               603           LD A,(HL):INC (HL)
54E7  D9                  604           EXX
```

▶12月号の72~73ページに「日ペンの美子ちゃん」と書いてありましたが，いったい何者なんですか，美子ちゃんて。

松上　明正　(15)　鹿児島県

```
54E8  61 6F CD 201E        605           LD H,C:LD L,A :CALL #LOC
54ED  26 00 2D 2D 19       606           LD H,0:DEC L,L:ADD HL,DE
54F2  7E CD 1FF4           607           LD A,(HL):CALL #PRINT
54F6  3E DB                608           LD A,'ロ'
54F8  C3 1FF4              609           JP #PRINT
                           610
                           611 ;////////////////////////////////////////
                           612
54FB  21 02 46 4C 49 43    613 RCMSG:    DB 33, 2:DB "FLICK" ,1
5501  4B 01
5503  21 05 52 4F 55 4E    614           DB 33, 5:DB "ROUND" ,1
5509  44 01
550B  22 08 53 54 45 50    615           DB 34, 8:DB  "STEP" ,1
5511  01
5512  21 0B 4B 45 59 3A    616           DB 33,11:DB "KEY:"  ,2,2
5518  02 02
551A  52 29 65 74 72 79    617                     DB "R)etry",2,2
5520  02 02
5522  4E 29 65 78 74 02    618                     DB "N)ext" ,2,2
5528  02
5529  42 29 61 63 6B 02    619                     DB "B)ack" ,2,2
552F  02
5530  53 29 6B 69 70 02    620                     DB "S)kip" ,2,2
5536  02
5537  51 29 75 69 74 00    621                     DB "Q)uit" ,0
                           622 CLRMSG:
553D  04 17 43 4C 45 41    623           DB 4,23 :DB "CLEAR. "
5543  52 2E 20
                           624 HITMSG:
5546  48 69 74 20 5B 53    625           DB "Hit [SPACE] key!",0
554C  50 41 43 45 5D 20
5552  6B 65 79 21 00
                           626           ;--------------------
                           627 MOVKEY:
5557  32 34 36 38 4B 4A    628           DB "2468KJLI"
555D  4C 49
555F  1F 1D 1C 1E          629           DB 1FH,1DH,1CH,1EH
                           630
5563  52 502E              631           DB 'R' :DW RETRY
5566  4E 508E              632           DB 'N' :DW NEXT
5569  42 508B              633           DB 'B' :DW BACK
556C  53 5097              634           DB 'S' :DW SKIP
556F  51 5368              635           DB 'Q' :DW QUIT
5572  00 0000              636           DB 0   :DW 0
5575  00                   637           DB 0
                           638 MOVDAT:
5576  FFF0 51AF            639           DW -16,UPCHR
557A  0001 518B            640           DW   1,RGTCHR
557E  FFFF 5197            641           DW  -1,LFTCHR
5582  0010 51A3            642           DW  16,DWNCHR
                           643           ;--------------------
                           644 MSGDAT:
5586  4025                 645           DW BUFFER+2+35
5588  66 6F 72 01          646           DB "for",1
558C  40E0                 647           DW BUFFER+14+210
558E  53 57 4F 52 44 00    648           DB "SWORD",0

     649           ;--------------------
     650 TTLDAT1:
     651           DB $07,$E0,$3F,$3F ,$07,$80,$27,$3C
     652           DB $07,$E7,$27,$3F ,$07,$E7,$3F,$3F
     653           DB $01,$E0,$3F,$0F ,$07,$E0,$3F,$3F
     654 TTLDAT2:
     655           DB $7C,$83,$8E,$22 ,$40,$81,$11,$24
     656           DB $78,$81,$10,$28 ,$40,$81,$10,$34
     657           DB $40,$81,$11,$22 ,$40,$F3,$8E,$22
     658 WAVDAT:
     659           DB $F0,$C8,$F4,$AA,$F0,$CC,$F0
     660           DB $0F,$33,$0F,$55,$0B,$37,$0F
     661           ;--------------------
     662 KEYDAT:
```

```
663           DB $28,$46,$52,$26,$18,$24,$18,$14,$12,$14,$12,$26
664           DB $12,$46,$28,$44,$18,$14,$12,$16,$12,$54,$12,$14
665           DB $18,$14,$18,$56,$32,$46,$18,$24,$12,$14,$18,$76
666           DB $18,$54,$12,$14,$68,$52,$66,$68,$64,0
667           ;--------------------
668 STGDAT:
669           DB $80,$01,$80,$01,$BE,$7D,$A0,$45,$A6,$45
670           DB $A2,$45,$BE,$7D,$80,$01,$80,$01,$80,$01
671           DB $00,$00,$00,$8B,$8D,$89,$00,$00,$00,$83,$22,$64
672
673           DB $80,$01,$80,$01,$86,$61,$84,$01,$84,$21
674           DB $84,$21,$87,$E1,$80,$01,$80,$01,$80,$01
675           DB $00,$00,$00,$82,$72,$62,$52,$42,$32,$34,$66,$84
676
677           DB $84,$21,$84,$21,$84,$21,$84,$21,$84,$01
678           DB $84,$21,$80,$21,$84,$21,$84,$21,$84,$21
679           DB $00,$00,$00,$00,$8E,$8D,$8C,$7C,$6C,$34,$AE,$AC
680
681           DB $80,$01,$9F,$F1,$80,$05,$AF,$E5,$E0,$07
682           DB $80,$01,$AF,$F5,$80,$01,$8F,$F1,$80,$01
683           DB $67,$63,$84,$58,$7C,$00,$00,$00,$32,$57,$68,$6A
684
685           DB $8F,$F1,$81,$81,$85,$A1,$80,$01,$F3,$CF
686           DB $F3,$CF,$80,$01,$85,$A1,$81,$81,$8F,$F1
687           DB $2D,$3D,$4D,$00,$00,$00,$82,$00,$92,$44,$AE,$AD
688
689           DB $CC,$33,$80,$01,$80,$01,$8F,$B1,$88,$01
690           DB $80,$11,$8D,$F1,$80,$01,$80,$01,$CC,$33
691           DB $64,$00,$00,$49,$00,$76,$5B,$5C,$5D,$67,$58,$72
692
693           DB $80,$01,$8D,$81,$90,$61,$90,$19,$A0,$05
694           DB $A0,$05,$98,$09,$86,$09,$81,$B1,$80,$01
695           DB $00,$00,$00,$00,$66,$67,$68,$69,$6A,$47,$A1,$46
696
697           DB $87,$E1,$80,$09,$94,$29,$80,$01,$F1,$8F
698           DB $F1,$8F,$80,$01,$94,$29,$80,$01,$83,$C1
699           DB $89,$39,$3B,$00,$00,$36,$00,$00,$86,$43,$AE,$4B
700
701           DB $A0,$01,$A0,$71,$BC,$01,$82,$85,$80,$45
702           DB $80,$05,$80,$85,$BC,$01,$A0,$71,$A0,$01
703           DB $00,$00,$27,$57,$25,$97,$67,$95,$71,$38,$11,$5A
704
705           DB $80,$07,$A1,$07,$A1,$01,$BB,$79,$80,$03
706           DB $83,$C1,$B8,$1D,$80,$01,$9C,$39,$83,$C1
707           DB $00,$2B,$25,$24,$00,$00,$87,$88,$89,$3A,$AE,$86
708
709           DB $A1,$85,$A1,$85,$80,$01,$8E,$11,$80,$11
710           DB $80,$11,$8E,$F1,$A8,$11,$AA,$91,$88,$11
711           DB $00,$00,$4A,$00,$9D,$8D,$6D,$5D,$4D,$8C,$AA,$49
712
713           DB $8F,$01,$89,$1D,$89,$01,$83,$71,$89,$41
714           DB $88,$15,$88,$85,$80,$85,$88,$95,$88,$81
715           DB $22,$32,$42,$52,$62,$72,$82,$92,$00,$8B,$11,$86
716
717           DB $88,$11,$A8,$15,$A0,$01,$A9,$91,$89,$91
718           DB $89,$91,$89,$95,$80,$05,$A8,$15,$88,$11
719           DB $00,$00,$00,$27,$28,$00,$97,$98,$99,$63,$6C,$96
720
721           DB $A0,$01,$83,$C1,$A3,$C1,$80,$41,$A2,$01
722           DB $83,$C1,$BB,$DD,$A0,$05,$A0,$05,$B9,$9D
723           DB $00,$00,$2B,$2C,$2D,$4B,$4C,$4D,$00,$5C,$A1,$3C
724
725           DB $80,$01,$BB,$D5,$A0,$01,$A3,$DF,$A3,$01
726           DB $A0,$01,$A3,$C1,$A2,$41,$BA,$01,$80,$01
727           DB $8A,$44,$00,$25,$69,$8C,$3B,$3A,$6B,$6A,$68,$8D
728
729           DB $81,$E3,$A9,$F9,$81,$FD,$A9,$65,$89,$25
730           DB $C0,$01,$C0,$29,$8F,$A9,$80,$01,$E2,$39
731           DB $96,$00,$32,$34,$00,$73,$75,$77,$79,$83,$97,$36
```

X1/X1turbo エンデューロレーサー

©SEGA

Matsuzaki Tsuyoshi
松崎 剛史

X68000「アルルの女」よりファランドール

Nagata Tohshi
永田 央

お待たせしました，Oh!X LIVE in '89のスタートです。今月はゲーム音楽とクラシックの2曲。皆さんの作品のおかげで，Oh!Xの守備範囲もすっかり幅広くなりました。では，今年もはりきっていきましょう。

まずはゲームミュージック

新春第1弾はSEGAのレーシングゲーム，エンデューロレーサーよりタイトルソングENDURO RACERのX1/X1turbo用プログラムです。

エンデューロレーサーは1986年に登場のゲームですが，あの筐体にはまり込んでモトクロス気分を楽しんだ人がたくさんいらっしゃることでしょうね。

作者の松崎さんはVIP ROOMのメンバーで，現在中学3年生。受験が終わってから投稿に励もうと考えていたそうですが，LIVE in '89の呼び声を聞き，とうとう乗せられてしまったとか。ほかにもいろいろなゲームミュージックを送ってくれたのですが，今回は，以前作ったものをバージョンアップしてみたというこの曲を採用させていただきました。

実行には祝版MMLおよびリスト1，リスト2が必要です。

なお，1130行と1140行のシングルクォーテーションを取ると，1988年5月号に掲載されたX-Keyboardのプログラムが使えるようになります。X-Keyboardをお持ちの方は試してみてください。

松崎さんの所属するVIP ROOMでは，Oh!X LIVE in '89を占領しよう，と怪気炎を上げているそうです。もちろん，そうは問屋が卸さんという方も多いので，'89はとても期待できそうですね。

お次はビゼー

第2弾はクラシック。ビゼー作曲「アルルの女」第2組曲より「ファランドール」です。アルルの女の中でも有名な曲のひとつですので，お馴染みの方も多いでしょう。

作者の永田さんは，昨年10月号に掲載されたバッハのアリアを作ってくれた人です。クラシック音楽の中にも楽しい曲がたくさんあることを知ってほしい，という彼の次回作も楽しみですね。

曲はX68000用OPMファイルの形で掲載します。テンポが速く，音符の数も多い曲で，ファイルもずいぶん大きなものになりましたが，がんばって聞いてみてください。

実行にはOPMDRV.Xが必要です。EDなどのエディタを使い，行番号とコロンを除いた部分を打ち込んでいってください。曲を聞くときは，

エンデューロレーサー

COPY faran.OPM OPM

のように，システム予約ファイルOPMにコピーして演奏します。

'89を盛り上げよう

X1勢に押され気味のMZですが，MZユーザーの力はこんなものではないはず。気合いを入れてがんばりましょう。

さて，早いもので，Oh!X LIVEが始まってから3年目に入りました。その間，投稿作品のレベルもどんどん上がってきています。先月号で発表したMusicBASICはもう打ち込みましたか？　この新バージョン対応の作品も楽しみに待っています。冬休みは思いきり音楽しましょう。

リスト1 ENDURO RACER音色定義

```
10 POKE &HAFDE,&HCA:POKE &HAFE1,0    ' "&" "&+" : "&+" "&"
20 POKE &HAE57,12, 0, 7, 0,25, 0,47, 0,20, 0,15  ' カウンタ カキカエ
30 'L26=80 , L27=L128 , L28=40 , L29=L5.25 , L30=L48 , L31=L64
40 '
50 '  &HAFDE &HAD74 , &HAFE1 &HAD76 , &HAE57 &HAE6A
60 '
70 MEM$(&HB190,36)=HEXCHR$("FC 00 4F 01 40 40 00 00 06 00 1F 1F
1F 1F 00 0F 10 07 40 00 00 C3 01 F7 96 A6 00 07 00 04 00 14 8A 0
A 00 80")  ' 1 SNARE DRUM
80 MEM$(&HB1B4,36)=HEXCHR$("FC 00 01 01 40 00 00 00 06 00 1F 1F
1F 1F 07 13 0E 0F 1F 1F 1F 1F CF CF DA DA 00 00 00 00 00 00 8A 0
A 00 00")  ' 2 BASS DRUM
90 MEM$(&HB1D8,36)=HEXCHR$("FE 00 00 00 00 00 04 00 00 00 1F 1F
1F 1F 01 10 0C 07 45 0B 06 09 25 A8 A6 87 00 00 00 00 00 00 80 0
0 00 00")  ' 3 TOM
100 MEM$(&HB1FC,36)=HEXCHR$("FC 00 0C 01 0F 0A 00 11 00 19 19 1B
 19 1F 05 91 05 11 C8 08 C8 88 61 F8 61 F8 00 00 00 00 00 00 80
00 00 00")  ' 4 C HI-HAT
110 MEM$(&HB220,36)=HEXCHR$("FC 00 0F 01 0A 0F 00 0D 00 19 1E 1F
 1E 1F 05 0C 07 0C C0 46 C2 86 02 57 12 57 00 00 00 00 00 00 80
00 00 00")  ' 5 O HI-HAT
120 MEM$(&HB244,36)=HEXCHR$("FC 00 0F 01 0A 0F 00 00 00 04 1E 1F
 1E 1F 00 06 06 08 C0 42 C2 82 00 46 20 86 00 00 00 00 00 00 80
00 00 00")  ' 6 CYMBAL
130 '
140 MEM$(&HB460,36)=HEXCHR$("FA 00 02 18 01 53 14 26 20 00 9F 9F
 5F 9F 0A 01 01 02 03 01 03 05 14 A6 04 15 00 00 00 00 00 00 80
00 00 00")  ' 21 KEYBORD
150 MEM$(&HB484,36)=HEXCHR$("C3 00 05 00 00 00 32 0B 0E 00 1F 1F
 1F 1F 02 02 02 82 06 06 06 06 36 35 36 38 06 04 00 00 00 C8 80
00 02 00")  ' 22 E BASS
160 MEM$(&HB4A8,36)=HEXCHR$("FA 00 02 18 01 53 14 26 20 00 9F 9F
 5F 9F 06 06 06 06 02 02 02 02 44 46 44 45 00 00 00 00 00 00 80
00 00 00")  ' 23 KEYBORD 2
170 MEM$(&HB4CC,36)=HEXCHR$("FA 00 02 18 01 53 14 26 20 00 9F 9F
 5F 9F 03 03 03 03 00 00 00 00 34 36 34 35 00 00 00 00 00 00 80
00 00 00")  ' 24 KEYBORD 3
180 '
190 RUN "ENDURO RACER .MML"
200 '   ～～  ＥＮＤＵＲＯ  ＲＡＣＥＲ   ～～  （ｃ）ＳＥＧＡ
210 '
220 '  ＰＲＯＧＲＡＭＭＥＤ
230 '          ＶＩＰＲＯＯＭ  Ｎｏ．０２５    松崎  剛史
240 '
250 '   プログラムを入力する前に、ＭＭＬを拡張して下さい。
260 '
270 '
280 '  ＶＩＰＲＯＯＭに入りませんか？
```

リスト2 ENDURO RACER.MML

```
10 GOSUB 20:GOSUB 280:GOTO 570
20 T2$="G&G-&F&E":T3$="E&E-&D&D-":T4$="C&<B&B-&A>
30 Y1$="I2DRDI1BRI2DRRDI1BGG":X1$="I4CRCCRCCRCCRC
40 Y2$="I1BRI2DI1BI2DDRI1BI2DI1BI2DI1B":X2$="I4CCRCCRCI5CI4CI5CI
4CI5C
50 Y3$="I1BRI2DI1BI2DDRDDDDD
60 RETURN
70 A1$=">C+8.&C+8<B4A8G+&G+2C+E8F+G+2A8.BA8.G+8F+8.G+A8B
80 BA$="O2D8.>DR<D8RD>D<D>D<E8.>ER<E8RE>E8.<E+8.>E+8<E+>G+8<E+>B
8>C+<<F+8.>F+RE4<F+8>F+<
90 CA$="O4I23V111A4.>D4.D4.R8<B>C+RDC+4.<B4.A4.R8AARG+
100 DA$="O4I23V111F+4.A4.B4.R8GGRGG+4.G+4.E4.R8F+F+RE
110 GA$="O4I23V111D4.F+4.G+4.R8EEREF4.F4.
120 CD$="A4.>D4.D4.R8<B>C+RDC+4.D4.E8.D8.C+8AARG+
130 DD$="F+4.A4.B4.R8GGRGG+4.G+4.A8.G+8.G+8F+F+RE
140 GD$="D4.F+4.G+4.R8EEREF4.F4.G+8.F+8.F+8
150 BB$="D8.>DR<D8RD>D<D>D<E8.>ER<E8RE>EFE<F+R>F+R8<F+F+R8>F+RF+
R8.<F+8.>F+ED<B>CC+
160 CB$="R4.>D4.R4.E4.C+RC+R4.C+RC+R4.>C+<
170 DB$="F+4.F+4.G+4.G+4.ARAR4.ARAR4.>A<
180 GB$="D4.D4.E4.E4.G+RG+R4.G+RG+R4.>F+<
190 EB$=Y1$+Y1$+"I2DRL31I3O4"+T2$+"R8."+T4$+"R16I2O1D16R8O4I3"+T
2$+"R8."+T2$+"R16I2O1D16O4I3"+T2$+T2$+T3$+T4$+T4$+"I2O1L16D
200 FB$=X1$+X1$+"CRV127I1BV119I4CRCV127I1BRV119I4CCRV127I1BV119I
4CRCI3O4L31"+T3$+"O1V119I4L16RCRRRRRC
210 BC$="<F+RF+R8>F+F+R8<F+RF+R8>F+F+R8ARB4
220 CC$="C+RC+R4.>C+RC+R2E4
230 DC$="ARAR4.>ARAR2A4
240 GC$="G+RG+R4.>G+RG+R2G+4
250 EC$="I2O1DRL31I3O4"+T2$+"R8.I1O1B16O4I3"+T3$+"O1I2D16R8I3O4"
+T2$+"R8."+T2$+"O1L16RRI1BRBRRR
260 FC$="CRI1BI4CRCRCCCRI1BI4CRCI3O4L31"+T3$+"R8L16O1I4CCCCRC
270 RETURN
280 A0$="O6I21L32V117T150
290 B0$="O2I22L16V119
300 E0$="O1I1L16V127
310 F0$="O1I6L16V119
320 Z1$="F+R>F+<":Z2$="ER>E<
330 EI$="I1BBBBBBBBBBBBB32B32BBI3O4L31"+T2$+T2$+T2$+T3$+T3$+T3$+
T4$+T4$+T4$
340 AI$="R2.>C+<BAGF+EDC+<BAGF+EDC+<B4":HI$="R31"+AI$:D1$="L8F+2
.E4.D4&D16E.F+.
350 A1$="L8>C+2.<B4.A4&A16B.>C+.<E16EF+.E4E.EF+4":E1$="O1L16"+ST
RING$(4,Y1$)
360 C1$="L8A2.G+4.F+4&F+16G+.A.C+16C+R.C+4C+.C+&C+4":F1$=STRING$
(4,X1$)
370 B1$=STRING$(4,Z1$)+STRING$(4,Z2$)+Z1$+Z1$+Z1$+"F+R>A8RA<F+R>
G+<F+R>F+<F+R>E<
380 A2$=">C+2.<B2>E.E.F+.<E16EF+.E4F+.AF+4":D2$="F+2.E2G+.G+.A.
390 C2$="A2.G+2B.B.>C+.<C+16C+RC+4C+.C+.
400 H4$=">C+2.<B2>E.E.F+.<E16EF+.E4E.AR16B
410 B2$=STRING$(4,Z1$)+STRING$(4,Z2$)+Z1$+Z1$+Z1$+"F+R>E8R<F+>C+
R<F+BRF+>CRC+<
420 B3$=STRING$(4,Z1$)+STRING$(4,Z2$)+STRING$(8,Z1$)
430 A3$=">C+2.<B4.A4&A16B.>C+.<E16EF+.A4G+.F+.EF+16
440 C3$="A2.G+4.F+4&F+16G+.A.C+16C+R.C+4C+.C+.C+.
450 B4$=STRING$(4,Z1$)+STRING$(4,Z2$)+Z1$+Z1$+Z1$+"F+R>E8RF+<F+R
>G+<F+R>A<F+R>G+<
460 A4$=">C+2.<B2>E.E.F+.<E16EF+.E4E.EF+.":D4$="F+2.E2G+.G+.A.R1
6R2.DR16E
470 C4$="A2.G+2B.B.>C+.<C+16C+R.C+4C+.C+C+":G4$="A2.G+2B.B.>C+.<
C+16C+RC+4F+R16G+
480 B5$=STRING$(4,Z1$)+STRING$(4,Z2$)+Z1$+Z1$+Z1$+"F+R>G+ARF+<F+
R>E<F+R>C+<F+R>E<
490 B6$=STRING$(4,Z1$)+STRING$(4,Z2$)+STRING$(4,Z1$)+"F+RB>CRC+E
REF+RA
500 A7$="L16>C+8C+<BBBA8AG+8F+>C+8C+<BBBA8G+8.F+":C7$="V114"+A7$
510 A8$=">C+8C+<BBBA8A>E8F+&F+2."+STRING$(7,"E")+"8":A7$="Y51,65
"+A7$
520 C8$=">C+8C+<BBBA8AE8F+&F+2."+STRING$(7,"G+")+"8
530 B8$=">C+8C+<BBBA8AE8F+&F+2."+STRING$(7,"E")+"8
540 E7$="I3O4L31"+T2$+"I1O1B16O4I3"+T2$+T3$+T3$+T3$+T4$+"R16"+T4
$+"I1O1L16BBBL31O4I3"+T2$+"I1O1B16O4I3"+T2$+"E&E-E&E-"+T3$+T3$+T
4$+"R16"+T4$+"R8I1O1B16
550 E8$="O4L31I3"+T2$+"I1O1B16O4I3"+T2$+T3$+T3$+T3$+"I1O1L16B8B8
R8"+Y1$+"BBBBBBBR2B8.
560 F7$="R8CRCRCRCRCRCRCRCRCCRC":F8$="CRCRCRCCCC":RETURN
570 PLAY 0
580 PLAY A0$+":"+B0$+":"+A0$+":"+A0$+":"+E0$+":"+F0$+":"+A0$+":"
+A0$
590 PLAY AI$+"::K5R32"+AI$+":K8"+AI$+":"+EI$+":";PLAY ":K15"+HI$
+"K10:K10"+HI$
600 PLAY "::::::K10:K10
610 PLAY A1$+":"+B1$+":"+C1$+":"+D1$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+C1$+":"+A1$
620 PLAY A2$+":"+B2$+":"+C2$+":"+D2$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+C2$+":"+A2$
630 PLAY A3$+":"+B3$+":"+C3$+":"+D1$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+C3$+":"+A3$
640 PLAY A4$+":"+B4$+":"+C4$+":"+D4$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+G4$+":"+H4$
650 PLAY A3$+":"+B5$+":"+C3$+":"+D1$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+C3$+":"+A3$
660 PLAY A4$+":"+B6$+":"+C4$+":"+D2$;PLAY ":"+E1$+":"+F1$;PLAY "
:"+C4$+":"+A4$
670 PLAY A7$+":"+A7$+":"+C7$+":"+A7$;PLAY ":"+E7$+":"+F7$;PLAY "
:"+C7$+":"+A7$
680 PLAY A8$+":"+B8$+":"+C8$+":"+A8$;PLAY ":"+E8$+":"+F8$;PLAY "
:"+C8$+":"+A8$
690 CLEAR:GOSUB 20
700 FOR I=1 TO 2
710 A$="Y55,40O5L16C+1&C+8<G+8A8B8G+2&G+8.E2.F+&
720 B$="O2F+RR>F+RRERRF+RR<F+RF+>F+<F+>F+E8RF+8R<ERR>ERRDRRERR<E
RE>E<E>ED8RE8R<
730 C$="O5I23V111L16R8.F+RRE8 F+RRF+8RF+F+RRE8 F+R4. E RRD+8E RR
E8 RE E RRD+8E
740 D$="O5I23V111L16R8.C+RRC+8C+RRC+8RC+C+RRC+8C+R4.<B RRB8 B RR
B8 RB B RRB8 B
750 G$="O4I23V111L16R8.A RRA8 A RRA8 RA A RRA8 A R4. G+RRG+8G+RR
G+8RG+G+RRG+8G+
760 E$=STRING$(4,Y1$):F$=STRING$(4,X1$):B$="V114"+B$
770 "P"
780 A$="F+2&F+8.>C+&C+2<B8.A>E1.
790 "P"
800 A$="F+2.&L8F+G+ABAG+L16E2&E8.<B2.>E&
810 B$="F+RR>F+RRERF+8RR<F+RF+>F+<F+>F+E8RF+8R<ERR>ERRDRRERR<ERE
>E<E>ED8RE8R<
820 "P"
830 A$="E2.&E8.C+8.<B8AG+8AG+8.&G+8F+&F+2.F+8G+A8B
840 B$="ERR>ERRDRRERR<ERE>E<E>ED8RE8R<F+8.>F+RRE8.F+R<F+RRF+>F+<
F+>F+<F+R>F+<F+R>F+<
850 C$="R8.E RRD+8E RRE8 RE E RRD+8 E RRRF+RRE8 F+RRF+8RF+F+RRE8
 F+RR D
860 D$="R8.B RRB8 B RRB8 RB B RRB8  B>RRRC+RRC+8C+RRC+8RC+C+RRC+
8C+RR<B
870 G$="R8.G+RRG+8G+RRG+8RG+G+RRG+8G+ RRRA RRA8 A RRA8 RA A RRA8
 A RR G+
880 E$=Y1$+Y1$+Y1$+"I2DRDR8DI1BGGBGG
890 "P"
900 GOSUB 70
910 A$=A1$:E$=STRING$(4,Y1$):B$=BA$:C$=CA$:D$=DA$:G$=GA$
920 "P"
930 A$=">C+8.&C+8<B4A8.G+2&G+A8.G+8F+1F+8G+A8B
940 C$=CD$:D$=DD$:G$=GD$
950 "P"
960 A$=A1$
970 C$=CA$:D$=DA$:G$=GA$
980 "P"
990 A$=">C+8.&C+8<B4A8.>E2C+G+8F+2.&F+8
1000 B$=BB$:C$=CB$:D$=DB$:E$=EB$:F$=FB$:G$=GB$
1010 IF I=1 THEN E$=Y1$+Y1$+Y1$+"I2DRDI1BGGI2DRI1B":F$=X1$+X1$+X
1$+"CRCCRCCRC
1020 "P":NEXT
1030 P$="O7L32C+<BAGF+EDC+<BAGF+EDC+<BAGF+ED16&D
1040 A$="R2.R2.R2."+P$
1050 B$=BC$
1060 C$=CC$+"R2.I21K15R31V101"+P$+"K5V111
1070 D$=DC$+"R2.I21K0R32V103"+LEFT$(P$,LEN(P$)-1)+"K5V111
1080 G$=GC$+"R2.I21K8V106"+P$+"K5V111
1090 E$=EC$+"I2DRRDRRDRRDRRI1BBBI3O4L31"+T2$+T2$+T2$+T3$+T3$+T3$
+T4$+T4$+T4$+"L16O1
1100 F$=FC$+"CRRCRRCRRCRR
1110 "P"
1120 A$="Y55,40R1.R1.
1130 'KEY0,CHR$(11)+"DELETE 580,1130"+CHR$(13,11)+"G.1140"+CHR$(
5,13) : END
1140 'GOSUB 20:GOSUB 80
1150 B$="O2F+RF+>F+<F+>E<F+>C+<F+BF+AF+RF+>F+<F+>E<F+>A<F+>G+<F+
>F+<F+RF+>F+<F+>EC+<F+BF+AF+>F+<F+>E<F+>G+<F+>F+<F+>A<F+>G+<
1160 C1$="O5L16C+R8.R1D+4C+R1R<A RRB ":C$=C1$
1170 D1$="O4L16A R8.R1A4 A R1R F+RRG+":D$=D1$
1180 G1$="O4L16F+R8.R1F+4F+R1R D RRE ":G$=G1$
1190 E1$=Y2$+Y2$+Y2$+"B32B32BBBI3O4L31"+T2$+T2$+T2$+T3$+T3$+T3$+
T4$+T4$+"O1L16
1200 F1$=X2$+X2$+X2$+"R4.RI5CI4CI5CI4CI5C
1210 E$=E1$:F$=F1$
1220 "P"
1230 B$="F+RF+>F+<F+>E<F+>C+<F+BF+AF+RF+>F+<F+>E<F+>A<F+>G+<F+>F
+<F+RF+>F+<F+>EF+<F+>G+<F+>A<F+>B<F+>>C<<F+>>C+<<F+>B<F+>A<F+>F+
<F+
1240 E2$=Y2$+Y2$+Y2$+"BI2DDL31O4I3"+T2$+T2$+"O1I2D16O4I3"+T3$+T3
$+"O1I2D16O4I3"+T4$+"O1I2L16DD
1250 F2$=X2$+X2$+X2$+"R8I4CRCRCI5CI4CRCI5C
1260 E$=E2$:F$=F2$
1270 "P"
1280 B_$="F+B>CC+EF+<":B$=B_$+B_$+B_$+"F+R>A<F+R>F+<"+B_$+B_$+B_
$+">EF+AF+AB<
1290 E3$=Y3$+Y3$+Y3$+"I3O4L31"+T2$+T2$+T2$+T2$+T3$+T3$+T3$+T4$+T
4$+T4$+T4$+T4$+"O1L16
1300 F3$=X2$+X2$+X2$+"I4CRRCRRCI5CI4CCI5CI4C
1310 E$=E3$:F$=F3$
1320 "P"
1330 C2$="O5C+R8.R1D+4C+R2R8.E E E E E E E R F+4":C$=C2$
1340 D2$="O4A R8.R1A4 A R2R8.B B B B B B B R>C+4":D$=D2$
1350 G2$="O4F+R8.R1F+4F+R2R8.G+G+G+G+G+G+G+R A4 ":G$=G2$
1360 B$=B_$+B_$+B_$+">EC+EC+<B>C+<"+B_$+">AF+ABRAB>C+<B>C+EC+8RF
+4
1370 E4$=Y3$+Y3$+"I1B32B32BBL31O4I3"+T2$+T2$+T2$+T3$+T3$+T3$+T4$
+T4$+T4$+"O1L16I1BBBBBBB
1380 F4$=X2$+X2$+"R2.I4CRRCRRR8I6C
1390 E$=E4$:F$=F4$
1400 "P"
1410 A$="AB>E<":A$="O4L16"+A$+A$+A$+A$+">C+<B>E<B>F+<B>A<B>G+<B>
F+<B"+A$+A$+A$+A$+"AB>C+EF+EC+<BABAG+
1420 C$=C1$:D$=D1$:G$=G1$
1430 B$="O2F+8.>F+R<F+>E8.F+8.<F+RF+>F+<F+>E4F+8.<F+8.>F+R<F+>E8
.F+8.<F+RF+>F+<F+>F+E8.F+8.
1440 E$=STRING$(4,Y1$):F$=STRING$(4,X1$)
1450 "P"
1460 A$="F+4EF+C+EF+EF+AF+ABAB>C+<B>C+EF+8EEC+<B>C<BABAF+AF+EF+A
BAB>C+<B>C+EF+8L32C+E
1470 "P"
1480 A$=STRING$(8,"A8F+C+")+"L16F+FEEF+AF+ABAB>CC+EC+EEEERF+F+8E
1490 B$="O2F+8.>F+R<F+>E8.F+8.<F+RF+>F+<F+>E4F+8.<F+8.>F+R<F+>E8
```


▶“営業努力”をする歯医者さんはたくさんいることでしょうが，でも，うちの近所の歯医者さんは診療台を降りようとした瞬間に詰め物が取れてしまうほどの努力家です。教訓「過ぎたるは及ばざるがごとし」。
水野　圭　(17)　愛知県

```
.F+R<F+"+STRING$(6,">E<E")
1500 E5$=Y1$+Y1$+"I2DRI3O4L31"+T2$+"I1O1L16BRI2DI3O4L31"+T3$+"L1
6O1RI1BRRB"+Y1$
1510 F5$=X1$+X1$+"CRO4I3L31"+T3$+"I4O1L16CRCO4I3L31"+T4$+"L16RI4
O1CCRC"+X1$
1520 E$=E5$:F$=F5$
1530 "P"
1540 A$="A8F+32E32C+8F+8EC+<B>CC+<BAF+BAF+AF+E8F+8EC+<B>C+<BABAF
+AF+E32F+32>EEEEEEERF+4
1550 C$=C2$:D$=D2$:G$=G2$
1560 B$="O2F+8.>F+R<F+>E8.F+8.<F+RF+>F+RE8R<F+>F+8.<F+8.>F+R<F+>
ER<F+>F+8.F+<F+>F+<F+>F+<F+>F+R>C+4
1570 E$=Y1$+Y1$+"I2DRDI1BI2DDDRDI1BRI2DI1BBBBBBB
1580 F$=X1$+X1$+"CRCRI5CI4CCRCRCRCRRCRRRRI6C
1590 "P"
1600 A$="R1.R1.
1610 B$="O2F+8.>F+<R1E4F+R8.R1E4
1620 C$="O5C+R8.R1D+4C+R2R8.>L8V119C+C+C+C+C+V111L16
1630 D$="O4A R8.R1A4 A R2R8.>L8V119A A A A A V111L16
1640 G$="O4F+R8.R1F+4F+R2R8.>L8V119F+F+F+F+F+V111L16
1650 E$=E1$:F$=F1$
1660 "P"
1670 B$="F+R8.R1E4F+R1RE4.
1680 C$="O5C+R8.R1D+4C+R2.R>V119C+V111
1690 D$="O4A R8.R1A4 A R2.R>V119A R8.G8 A V111
1700 G$="O4F+R8.R1F+4F+R2.R>V119F+R8.F+8F+V111
1710 E$=E2$:F$=F2$
1720 "P"
1730 C$=C1$:D$=D1$:G$=G1$
1740 B$="F+R8.R1E4F+
1750 E$=E3$:F$=F3$
1760 "P"
1770 C$=C2$:D$=D2$:G$=G2$
1780 B$="F+R1R8.E4F+R2R8.BBBBBBBR>F+4
1790 E$=E4$:F$=F4$
1800 "P"
1810 A$="O6L16I24Y55,40C+1.<C+2.&C+2E8.F+&
1820 B$="O2F+8.>F+R<F+>E8.F+8.<F+RF+>F+<F+>E4F+8.<F+8.>F+R<F+>ER
<F+>F+8.<F+RF+>F+<F+>F+E8.F+8.
1830 C$=C1$:D$=D1$:G$=G1$
1840 E$=STRING$(4,Y1$):F$=STRING$(4,X1$)
1850 "P"
1860 A$="L8F+FF+CE<B>E<B>D+<A+>D<A>C+<G+>C<GBAF+2.
1870 B$="O2F+8.>F+R<F+>E8.F+8.<F+RF+>F+<F+>E4F+8.<F+8.>F+R<F+>E8
.F+8.<F+RF+>A8.B8.>C+8.
1880 C$="O5C+R4C+R4RC+R2D+4C+R4C+R4RC+R4R<A4B
1890 D$="O4A R4A R4RA R2A4 A R4A R4RA R4R F+4G+
1900 G$="O4F+R4F+R4RF+R2F+4F+R4F+R4RF+R4R D4 E
1910 "P"
1920 A$="AB>C+<BAF+":A$="L16"+STRING$(4,A$)+"L29"+A$+STRING$(2,"
AB>C+<BF+")+"A2&A8&L32AB>C+C+
1930 B$="O2F+8.>F+8.E8.ER<F+R8F+>F+8RE8.F+8.<F+8.>F+8.E8.F+R<F+"
+STRING$(6,">E<E")
1940 C$="O5C+R4C+R4RC+R2D+4C+R4C+R8.C+8C+R4.<A RRB
1950 D$="O4A R4A R4RA R2A4 A R4A R8.A8 A R4. F+RRG+
1960 G$="O4F+R4F+R4RF+R2F+4F+R4F+R8.F+4F+R4. D RRE
1970 E$=E5$
1980 "P"
1990 A$="E8.&E8C+EF+4&F+C+EF+L16AF+EC+EF+A4&AF+>C+8<BAF+EL29F+AF
+EC+<B>CC+<BA>C+<BEF+AF+L16ERF+4>
2000 B$="O2F+8.>F+R<F+>E8.F+R<F+R8F+>F+<F+>E4F+8.<F+8.>F+8.E8.F+
8.<F+R>C+<BABR8>F+4<
2010 C$="O5C+R4C+R4RC+R2D+4C+R4C+R4RC+E E E E E E R8 F+4
2020 D$="O4A R4A R4RA R2A4 A R4A R4RA B B B B B B R8>C+4
2030 G$="O4F+R4F+R4RF+R2F+4F+R4F+R4RF+G+G+G+G+G+G+R8 A4
2040 E$=Y1$+Y1$+"I2DRDI1BI2DDDRDI1BRI2DI1BBBBBBB
2050 F$=X1$+X1$+"CRCRI5CI4CCRCRCRCRRCRRR8.I6C
2060 "P"
2070 AA$="I21Y55,40C+2<B8.AG+2C+E8F+G+2A8.BG+4.F+8G+A8B
2080 A$=AA$:B$=BA$:C$=CA$:D$=DA$:G$=GA$
2090 E$=STRING$(4,Y1$):F$=STRING$(4,X1$)
2100 "P"
2110 A$=">C+8&C+8.<B4A8.G+2A4G+8F+&F+2>F+&F+2G+A8B>
2120 C$=CD$:D$=DD$:G$=GD$
2130 "P"
2140 A$=AA$
2150 C$=CA$:D$=DA$:G$=GA$
2160 "P"
2170 A$=">C+8&C+8.<B4A8.G+2A8.BA8.G+8.F+2L8>C+<BAG+F+
2180 B$=BB$:C$=CB$:D$=DB$:G$=GB$
2190 E$=Y1$+Y1$+Y1$+STRING$(12,"B"):F$=X1$+X1$+X1$
2200 "P"
2210 A$="L16F+4.F+8C+C+8EA8.G+8.F+8.G+8>C+&C+2<BA8G+4F8F+8.G+8.A
2220 B$=BA$:C$=CA$:D$=DA$
2230 E$=STRING$(4,Y1$):F$=STRING$(4,X1$)
2240 "P"
2250 A$="B8AG+8F+8.G+8.AB8>C+<B8A8.G+8.EF+2.>C+<BA>C+<BA>C+<BABR
>F+&
2260 C$=CD$:D$=DD$:G$=GD$
2270 "P"
2280 A$="F+2C+E8F+&F+4.A4.G+4.F4.F+2E8.C+
2290 C$=CA$:D$=DA$:G$=GA$
2300 "P"
2310 A$="<B8>CC+8<A8.F+8.A>C+8<BA8B8.G+8.AE4&EEF+2.&F+8A8.>F+&
2320 B$=BB$:C$=CB$:D$=DB$:E$=EB$:F$=FB$:G$=GB$
2330 "P"
2340 A$="F+1.
2350 B$=BC$:C$=CC$:D$=DC$:E$=EC$:F$=FC$:G$=GC$
2360 "P"
2370 CLEAR:GOSUB 20:GOSUB 280
2380 PLAY "V117:O2V119:V117I21:V117I21:::V117I21K10:V117K10
2390 A1$="O5L8C+2.<B4.A4&A16B.>C+.E16EF+.E4E.EF+4":D1$="O4L8F+2.
E4.D4&D16E.F+.
2400 C1$="O4L8A2.G+4.F+4&F+16G+.A.>C+16C+R.C+4C+.C+&C+4<
2410 E1$=STRING$(4,Y1$):F1$=STRING$(4,X1$)
2420 A2$="C+2.<B2>E.E.F+.E16EF+.E4F+.AF+4":D2$="F+2.E2G+.G+.A.
2430 C2$="A2.G+2B.B.>C+.C+16C+RC+4C+.C+.<
2440 A3$="C+2.<B4.A4&A16B.>C+.E16EF+.A4G+.F+.EF+16
2450 C3$="A2.G+4.F+4&F+16G+.A.>C+16C+R.C+4C+.C+.C+.<
2460 A4$="C+2.<B2>E.E.F+":C4$="A2.G+2B.B.>C+":D4$="F+2.E2G+.G+.A
2470 B4$=STRING$(4,Z1$)+STRING$(4,Z2$)+Z1$
2480 E4$=Y1$+Y1$+"BRB":F4$=X1$+X1$+"CRC
2490 PLAY A1$+":"+B1$+":"+C1$+":"+D1$;PLAY ":"+E1$+":"+F1$;PLAY
":"+C1$+":"+A1$
2500 PLAY A2$+":"+B2$+":"+C2$+":"+D2$;PLAY ":"+E1$+":"+F1$;PLAY
":"+C2$+":"+A2$
2510 PLAY A3$+":"+B3$+":"+C3$+":"+D1$;PLAY ":"+E1$+":"+F1$;PLAY
":"+C3$+":"+A3$
2520 PLAY A4$+":"+B4$+":"+C4$+":"+D4$;PLAY ":"+E4$+":"+F4$;PLAY
":"+C4$+":"+A4$
2530 END
2540 LABEL"P"
2550 PLAY     A$;PLAY ":"+B$;PLAY ":"+C$;PLAY ":"+D$;
2560 PLAY ":"+E$;PLAY ":"+F$;PLAY ":"+G$;PLAY ":"+A$
2570 RETURN
```

リスト3 faran.OPM

```
==================  faran.opm  ===================
 1: /    G.Bizet, "l'Arlesienne" 2nd Suite
 2: /    no.4, Farandole
 3: /      for X68000 with opmdrv.x, Arr.by T.N.
 4: (i)
 5: (m1,8000)
 6: (a1,1)
 7: (m2,8000)
 8: (a2,2)
 9: (m3,8000)
10: (a3,3)
11: (m4,8000)
12: (a4,4)
13: (m5,8000)
14: (a5,5)
15: (m6,8000)
16: (a6,6)
17: (m7,8000)
18: (a7,7)
19: (m8,8000)
20: (a8,8)
21: / Sorry for too many timber data.
22: / timber number 1, 39, 40, 44 and 45 are from MP_SOUND.SND in
 SOUND pro-68K, SHARP.
23: (v40,0,61,15,2,0,203,14,0,4,0,3,83,31,7,0,3,1,29,0,1,0,0,1,14
,31,0,6,0,5,0,2,0,0,1,14,31,0,6,0,5,0,0,0,0,1,14,31,0,6,0,5,0,1,0
,0,1)
24: (v79,0,37,15,2,0,201,22,4,2,0,3,84,15,4,0,6,3,22,1,1,2,0,1,18
,3,0,4,4,0,2,4,1,0,1,18,1,0,4,7,0,2,1,5,0,1,18,3,0,4,0,0,2,1,0,0,
1)
25: (v1,0,62,15,2,0,205,40,4,3,1,3,80,25,10,15,9,5,34,0,2,0,0,1,1
8,0,0,9,4,11,0,2,0,0,1,17,4,0,15,1,28,0,2,0,0,1,16,9,0,15,0,7,0,2
,0,0,1)
26: (v94,0,60,15,2,0,204,74,7,0,0,3,67,31,8,0,9,6,33,1,2,5,0,1,18
,30,0,8,0,0,0,1,0,0,1,14,1,0,8,5,42,0,3,6,0,0,13,31,0,7,0,17,1,5,
2,0,1)
27: (v93,0,13,15,2,0,192,29,0,1,2,3,84,29,18,0,7,15,26,0,0,3,3,1,
31,15,18,10,3,0,0,0,2,1,1,31,25,12,7,0,0,0,0,5,2,1,31,25,12,6,0,0
,0,0,0,0,0)
28: (v39,0,58,15,2,0,203,42,0,4,0,3,83,27,5,0,7,1,34,0,2,1,0,1,28
,17,16,13,8,37,0,12,0,2,1,25,11,6,6,0,35,0,4,0,0,1,11,10,0,6,0,2,
2,2,0,0,1)
29: (v95,0,61,15,2,0,203,14,0,4,0,3,83,31,7,0,3,1,29,0,2,0,0,1,15
,31,0,7,0,0,0,4,0,0,1,15,31,0,7,0,0,0,1,0,0,1,15,31,0,7,0,0,0,2,0
,0,1)
30: (v96,0,58,15,2,0,198,36,7,2,1,3,83,31,14,0,1,7,26,2,2,1,0,1,2
8,5,0,5,5,34,2,6,0,0,1,29,24,1,4,6,10,1,1,0,0,1,17,3,0,6,0,0,1,2,
0,0,1)
31: (v99,0,62,15,2,0,200,58,11,2,1,3,70,31,15,1,5,9,33,1,1,3,0,1,
15,4,0,7,0,0,1,1,0,0,1,15,4,0,7,0,53,1,0,7,0,1,14,4,0,7,0,0,1,1,0
,0,1)
32: (v44,0,60,15,2,0,205,33,0,0,0,3,80,27,24,21,15,5,26,0,1,0,0,1
,31,13,0,8,15,0,0,1,0,0,1,31,31,0,15,3,25,0,1,0,0,1,31,16,0,6,15,
0,1,1,0,0,1)
33: (v45,0,62,15,2,0,205,29,4,2,1,3,80,31,19,0,9,15,32,0,1,0,0,1,
28,12,0,9,15,0,1,1,0,0,1,25,12,0,7,15,0,0,1,0,0,1,25,25,0,6,15,0,
0,0,0,2,1)
34: (v98,0,36,15,2,0,204,74,10,2,1,3,79,31,15,0,8,12,50,1,5,0,1,1
,24,1,0,8,5,12,1,3,0,0,1,21,4,0,8,0,29,1,1,0,0,1,19,0,0,8,5,0,1,2
,0,0,1)
35: (v101,0,61,15,2,0,203,14,0,4,0,3,83,31,7,0,3,1,29,0,2,0,0,1,1
5,31,0,8,0,0,0,4,0,0,1,15,31,0,8,0,0,0,1,0,0,1,15,31,0,8,0,0,0,2,
0,0,1)
36: (v100,0,44,15,2,0,205,0,0,3,0,3,72,14,8,0,8,3,29,0,1,0,0,1,17
```

▶次回の「言わせてくれなくちゃだワ」では，大学別の分類があってもいいのではないでしょうか。意外と面白い結果になるかもしれませんよ。受験雑誌(？)のOh!Xとしては，浪人生や現役の方へのサービスにもなるし（ならないって）。　渡辺　一矢　(19)　石川県

```
,6,0,8,3,3,0,1,0,0,1,16,8,0,8,3,35,0,1,0,0,1,16,5,0,8,2,3,0,1,0,0
,1)
37: (v97,0,61,15,2,0,205,27,0,3,0,3,66,15,10,0,5,2,32,0,1,0,0,1,2
1,31,0,7,0,0,0,2,0,0,1,21,31,0,7,0,0,0,2,0,0,1,21,31,0,7,0,0,0,3,
0,0,1)
38: (t1)T104@40V12Y48,9
39: (t2)@79V12Y49,9
40: (t3)@40V12Y50,9
41: (t4)@79V12Y51,5
42: (t5)@40V12Y52,5
43: (t6)@79V12Y53,5
44: (t7)@40V12Y54,5
45: (t8)@40V12Y55,0
46: (t1)O7V14L4R2T100Q6D>AT104Q7<D4.Q7R16E16Q6F8R16E16F8T98R16D16
Q8A4T104R8Q4F8Q6GA
47: (t2)O5V11L4R2Q6FEQ7F4.Q7R16B-16Q6F8R16A16A8R16B-16<Q8C4R8Q4>A
8Q6<CC
48: (t3)O6V12L4R2Q6AAQ7D4.Q7R16G16Q6F8R16A16F8R16F16Q8F4R8Q4A8Q6G
F
49: (t4)O5V13L4R2Q6D>AQ7<D4.Q7R16E16Q6F8R16E16F8R16D16Q8A4R8Q4F8Q
6GA
50: (t5)O5V12L4R2Q6FEQ7F4.Q7R16B-16Q6A8R16A16A8R16B-16<Q8C4R8Q4>A
8Q6<CC
51: (t6)O4V11L4R2Q6AAQ7D4.Q7R16G16Q6F8R16A16F8R16F16Q8F4R8Q4A8Q6G
F
52: (t7)O5V14L4R2Q6DCQ7>B-4.Q7R16<C16Q6D8R16C16D8R16>B-16Q8F4R8Q4
F8Q6EE-
53: (t8)O4V14L4R2Q6DCQ7>B-4.Q7R16<C16Q6D8R16C16D8R16>B-16Q8F4R8Q4
F8Q6EE-
54: (t1)Q5B-8Q4A8G8F8Q6EAQ4G8F8E8T96F8T100Q6D>A
55: (t2)Q5>B-8<Q4C8D8D8Q6DCCC+Q6>FE
56: (t3)Q5F8Q4E8D8F8Q6B-AQ4B-8A8G8A8Q6AA
57: (t4)Q5B-8Q4A8G8F8Q6EAQ4G8F8E8F8Q6D>A
58: (t5)Q5>B-8Q4<C8D8D8Q6DCCC+Q6>FE
59: (t6)Q5F8Q4E8D8F8Q6B-AQ4B-8A8G8A8Q6AA
60: (t7)Q5D8Q4C8>B-8A8Q6G<CFAQ6<DC
61: (t8)Q5D8Q4C8>B-8A8Q6G<CFAQ6<DC
62: (t1)T104Q7<D4.Q7R16E16Q6F8R16E16F8T98R16D16Q8A4T104R8Q4F8Q6GA
63: (t2)Q7F4.Q7R16B-16Q6F8R16A16A8R16B-16<Q8C4R8Q4>A8Q6<CC
64: (t3)Q7D4.Q7R16G16Q6F8R16A16F8R16F16Q8F4R8Q4A8Q6GF
65: (t4)Q7<D4.Q7R16E16Q6F8R16E16F8R16D16Q8A4R8Q4F8Q6GA
66: (t5)Q7F4.Q7R16B-16Q6A8R16A16A8R16B-16<Q8C4R8Q4>A8Q6<CC
67: (t6)Q7D4.Q7R16G16Q6F8R16A16F8R16F16Q8F4R8Q4A8Q6GF
68: (t7)Q7>B-4.Q7R16<C16Q6D8R16C16D8R16>B-16Q8F4R8Q4F8Q6EE-
69: (t8)Q7>B-4.Q7R16<C16Q6D8R16C16D8R16>B-16Q8F4R8Q4F8Q6EE-
70: (t1)Q5B-8Q4A8G8F8T98Q6FEQ8D4.R8Y48,5T100@40O6V13Q4D>A<
71: (t2)Q5>B-8<Q4C8D8D8Q6DC+Q8>A4.R8Y49,5@40O6V13Y49,15Q4D>A<
72: (t3)Q5F8Q4E8D8F8Q6B-GQ8F4.R8R2Y50,5
73: (t4)Q5B-8Q4A8G8F8Q6FEQ8D4.R8R2
74: (t5)Q5>B-8<Q4C8D8D8Q6DC+Q8>A4.R8R2
75: (t6)Q5F8Q4E8D8F8Q6B-GQ8F4.R8R2
76: (t7)Q5D8Q4C8>B-8A8Q6GAQ8<D4.R8R2
77: (t8)Q5D8Q4C8>B-8A8Q6GAQ8<D4.R8R2
78: (t1)T104Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16T106Q4
GA
79: (t2)Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16Q4GA
80: (t3)@40O5V13Q4D>A<Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R1
6F16
81: (t4)@40O5V13Y51,15Q4D>A<Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q
7A4.R16F16
82: (t5)R1R1
83: (t6)R1R1
84: (t7)R1R1
85: (t8)R1R1
86: (t1)Q3B-8A8G8F8Q4EAT104Q3G8F8E8F8T102Q4D>A<
87: (t2)Q3B-8A8G8F8Q4EAQ3G8F8E8F8Q4D>A<
88: (t3)Q4GAQ3B-8A8G8F8Q4EAQ3G8F8E8F8
89: (t4)Q4GAQ3B-8A8G8F8Q4EAQ3G8F8E8F8
90: (t5)R1R1
91: (t6)R1R1
92: (t7)R1R1
93: (t8)R1R1
94: (t1)T104Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16Q4GA
95: (t2)Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16Q4GA
96: (t3)Q4D>A<Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16
97: (t4)Q4D>A<Q7D4.R16Q7E32.G64&Q5F8R16E16F8R16D16Q7A4.R16F16
98: (t5)R1R1
99: (t6)R1R1
100: (t7)R1R1
101: (t8)R1R1
102: (t1)T102Q3B-8A8T96G8F8T86Q4V12FT70V11E
103: (t2)Q3B-8A8G8F8Q4V12FV11E
104: (t3)Q4GEV12A>V11A
105: (t4)Q4GEV12A>V11A
106: (t5)R1
107: (t6)R1
108: (t7)R1
109: (t8)R1
110: /#1
111: (t1)T168Q3V10D8R4.R2R2R4.@1V7O6Q3L8Y48,15D
112: (t2)Q3V10D8R4.Y49,5R2R2R4.@94V4O5Q3L8D
113: (t3)Q3<V10D8R4.R2R2R2
114: (t4)Q3V10D8R4.Y51,5R2R2R2
115: (t5)R2R2R2R2
116: (t6)R2R2R2R2
117: (t7)R2R2R2R2
118: (t8)@93V7O3Q4L8V9D\6DDDV9DV6DDDV9DV6DDDV9DV6DDD
119: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV8Q3D
120: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV5Q3D
121: (t3)R2R2R2R2R2R2R2R2
122: (t4)@39V5Q3L8O3ARRRARRRARRRARRRARRRARRRARRRARRR
123: (t5)@39V5Q3L8O3F+RRRDRRRGRRRF+RRRF+RRRDRRRGRRRF+RRR
124: (t6)@39V5Q3L8O3DRRR>BRRR<C+RRRDRRRDRRR>BRRR<C+RRRDRRR
125: (t7)@39V7Q3L8O2DRRRDRRRDRRRDRRRDRRRDRRRDRRRDRRR
126: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
V9DV6DDD
127: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4T162RV9Q3>AT168
128: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV6Q3>A
129: (t3)R2R2R2R2R2R2R2R2
130: (t4)V6ARRRARRRARRRARRRARRRARRRARRRARRR
131: (t5)V6F+RRRDRRRGRRRF+RRRF+RRRDRRRGRRRF+RRR
132: (t6)V6DRRR>BRRR<C+RRRDRRRDRRR>BRRR<C+RRRDRRR
133: (t7)V8DRRRDRRRDRRRDRRRDRRRDRRRDRRRDRRR
134: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
V9DV6DDD
135: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
136: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
137: (t3)@39V8Q4L64O3Q3B8R16Q4B&<C+&D&E&F+8R8Q3>A8<R16Q4A&B&<C+&D
&E8R8
138: (t3)>>Q3B8R16Q4B&<C+&D&E&F+8R8Q3>A8<R16Q4A&B&<C+&D&E8R8
139: (t4)@39V8Q4L64O3Y51,15Q3D8R16Q4{B&B&<C+&D&E&}16F+8R8R8.{A&A&
B&<C+&D&}16E8R8>>Q3D8R16Q4{B&B&<C+&D&E&}16F+8R8R8.{A&A&B&<C+&D&}1
6E8R8
140: (t5)V7F+RRRERRRF+RRRERRR
141: (t6)V7>BRRR<C+RRR>BRRR<C+RRR
142: (t7)V9DRRRDRRRDRRRDRRR
143: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
144: (t1)B<C+DC+>BAGF+GABGL16A&G&A&B&L8T164AV10AT168
145: (t2)B<C+DC+>BAGF+GABGL16A&G&A&B&L8AV7A
146: (t3)>>>Q3<B8>R16Q4B&<C+&D&E&F+8R8Q3B8R16Q4G&A&B&<C+&D8R8Q3>G
8<R16Q4E&F+&G&A&B8R8Q3>A8<R16Q4A&B&<C+&D&E8R8
147: (t4)>>Q3D8R16Q4>{B&B&<C+&D&E&}16F+8R8Q3D8R16Q4{G&G&A&B&<C+&}
16D8R8R8.{E&E&F+&G&A&}16B8R8Q3>C+8<R16Q4{A&A&B&<C+&D&}16E8R8
148: (t5)F+RRRGRRRDRRRERRR
149: (t6)>BRRRBRRRBRRRARRR
150: (t7)DRRRDRRRDRRRDRRR
151: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
152: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
153: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
154: (t3)V9Q3>>B8R16Q4B&<C+&D&E&F+8R8Q3>A8<R16Q4A&B&<C+&D&E8R8>>Q
3B8R16Q4B&<C+&D&E&F+8R8Q3>A8<R16Q4A&B&<C+&D&E8R8
155: (t4)V9Q3>>D8R16Q4{B&B&<C+&D&E&}16F+8R8R8.{A&A&B&<C+&D&}16E8R
8
156: (t4)>>Q3D8R16Q4{B&B&<C+&D&E&}16F+8R8R8.{A&A&B&<C+&D&}16E8R8
157: (t5)V8F+RRRERRRF+RRRERRR
158: (t6)V8BRRR<C+RRR>BRRR<C+RRR
159: (t7)V10DRRRDRRRDRRRDRRR
160: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
161: (t1)B<C+DC+>BAGF+GABGQ8A4T160RQ3<V11DT168
162: (t2)B<C+DC+>BAGF+GABGQ8A4RQ3<V7D
163: (t3)>>>Q3<B8>R16Q4B&<C+&D&E&F+8R8Q3B8R16Q4G&A&B&<C+&D8R8Q3>G
8<R16Q4E&F+&G&A&B8R8Q3>A8<R16Q4A&B&<C+&D&E8@1V9O5L8Q3Y50,10D
164: (t4)>>Q3D8R16Q4>{B&B&<C+&D&E&}16F+8R8Q3D8R16Q4{G&G&A&B&<C+&}
16D8R8R8.{E&E&F+&G&A&}16B8R8Q3>C+8<R16Q4{A&A&B&<C+&D&}16E8R8Y51,5
165: (t5)F+RRRGRRRDRRRERRR
166: (t6)>BRRRBRRRBRRR<C+RRR
167: (t7)DRRRDRRRDRRRDRRR
168: (t8)V9DV6DDDV9DV6DDDV9DV6DDDV9DV6DDD
169: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4T164RV12Q3DT168
170: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV8Q3D
171: (t3)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV10Q3D
172: (t4)@39Q3L8O3V10AV7AAAV10AV7AAAV10AV7AAAV10AV7AAAV10AV7AAAV1
0AV7AAAV10AV7AAAV10AV7AAA
173: (t5)V10F+V7F+F+F+V10DV7DDDV10GV7GGGV10F+V7F+F+F+V10F+V7F+F+F
+V10DV7DDDV10GV7GGGV10F+V7F+F+F+
174: (t6)V10DV7DDD>V10BV7BBB<V10C+V7C+C+C+V10DV7DDDV10DV7DDD>V10B
V7BBB<V10C+V7C+C+C+V10DV7DDD
175: (t7)V10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV1
0DV7DDDV10DV7DDD
176: (t8)V10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV1
0DV7DDDV10DV7DDD
177: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4T162RV13Q3>AT168
178: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV9Q3>A
179: (t3)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RV11Q3>A
180: (t4)V11AV8AAAV11AV8AAAV11AV8AAAV11AV8AAAV11AV8AAAV11AV8AAAV1
1AV8AAAV11AV8AAA
181: (t5)V11F+V8F+F+F+V11DV8DDDV11GV8GGGV11F+V8F+F+F+V11F+V8F+F+F
+V11DV8DDDV11GV8GGGV11F+V8F+F+F+
182: (t6)V11DV8DDD>V11BV8BBB<V11C+V8C+C+C+V11DV8DDDV11DV8DDD>V11B
V8BBB<V11C+V8C+C+C+V11DV8DDD
183: (t7)V11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV1
1DV8DDDV11D4R4
184: (t8)V11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDDV1
1DV8DDDV11DV8DDD
185: /#2
186: (t1)T168@1O5L8Q3V13
187: (t2)T168@94O4L8Q3V9
188: (t3)T168@1O4L8Q3V11
189: (t4)T168@39
190: (t5)T168@39
191: (t6)T168@39O3L8Q3
192: (t7)T168@39O2L8Q3
193: (t8)T168@93O3L8Q4
194: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
195: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
196: (t3)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
197: (t4)V12L64O3Q3B8Q6B16Q4B&<C+&D&E&F+8>Q3B8A8Q6A16<Q4A&B&<C+&D
&E8>>Q3A8B8Q6B16Q4B&<C+&D&E&F+8>Q3B8A8Q6A16<Q4A&B&<C+&D&E8>>Q3A8
198: (t5)Q4L64Y51,15O2V13D8V12O3R16Q4{B&B&<C+&D&E&}16F+8R8O2V13D8
V12O4R16{A&A&B&<C+&D&}16E8R8O2V13D8V12O3R16Q4{B&B&<C+&D&E&}16F+8R
```

▶うちの親がワープロを買ったからといって，初めて使うのに年賀状を書こうとしている。プリントゴッコかなにかのほうが，絶対に早いのに……。　増田　拓史　(18)　愛知県

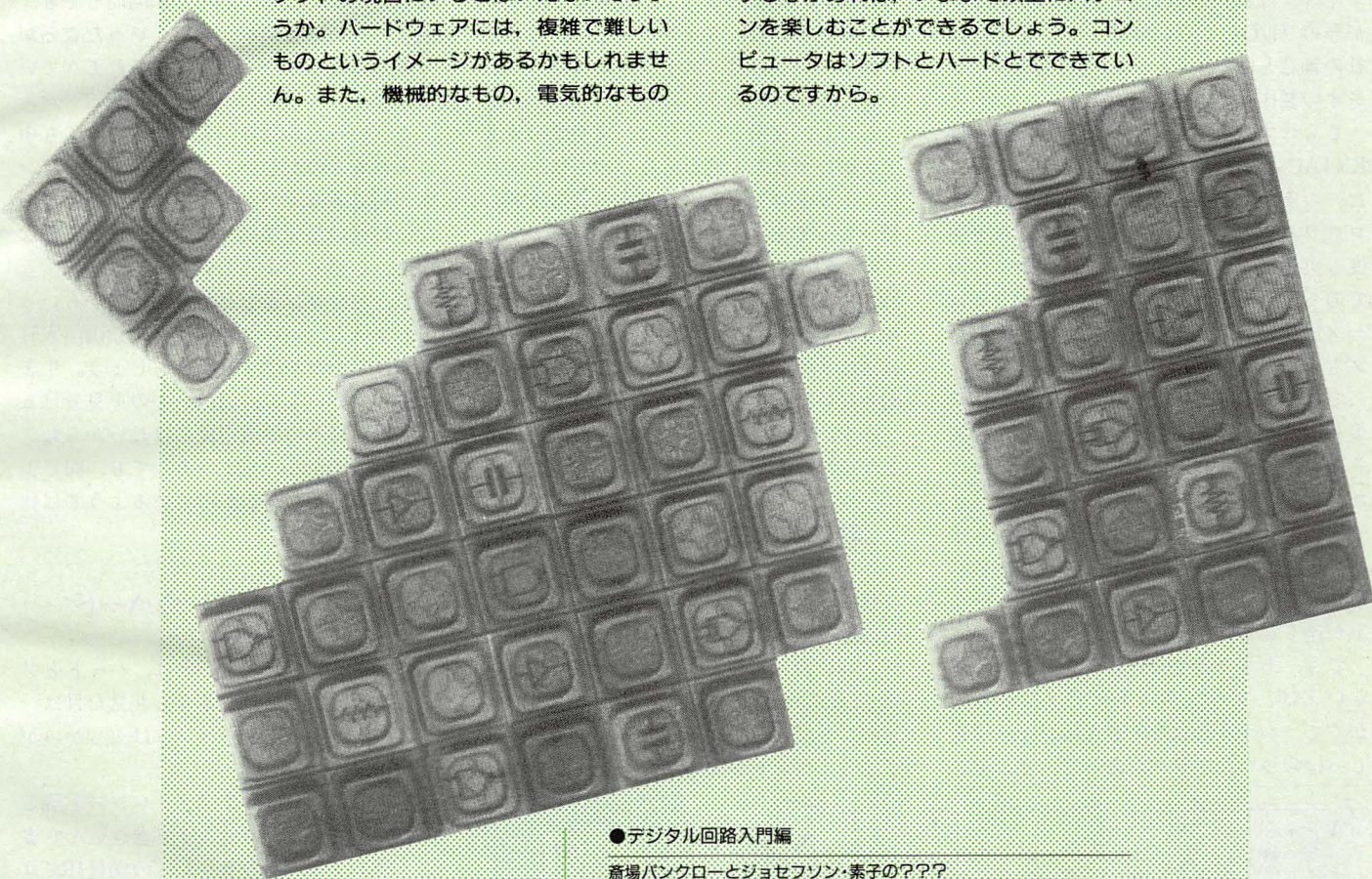
```
802V13D8V1204R16{A&A&B&<C+&D&}16E8R8
199: (t6)V12F+V9F+F+F+V12EV9EEEV12F+V9F+F+F+V12EV9EEE
200: (t7)V12BV9BBBV12<C+V9C+C+C+V12>BV9BBBV12<C+V9C+C+C+
201: (t8)V12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDD
202: (t1)B<C+DC+>BAGF+GABGL16A&G&A&B&L8T164AV14AT168
203: (t2)B<C+DC+>BAGF+GABGL16A&G&A&B&L8AV10A
204: (t3)B<C+DC+>BAGF+GABGL16A&G&A&B&L8AV12A
205: (t4)B8Q6B16>Q4B&<C+&D&E&F+8Q3B8B8Q6B16Q4G&A&B&<C+&D8>Q3B8G8Q
6G16<Q4E&F+&G&A&B8>Q3G8A8Q6A16<Q4A&B&<C+&D&E8>>Q3A8
206: (t5)O2V13D8V12O2R16{B&B&<C+&D&E&}16F+8R8O2V13D8V12O3R16{G&G&
A&B&<C+&}16D8R8O2V13D8V12O4R16{E&E&F+&G&A&}16B8R8O2V13D8V12O4R16{
A&A&B&<C+&D&}16E8R8
207: (t6)V12F+V9F+F+F+V12GV9GGGV12EV9EEEV12EV9EEE
208: (t7)V12>BV9BBBV12BV9BBBV12BV9BBBV12<C+V9C+C+C+
209: (t8)V12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDD
210: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
211: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
212: (t3)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
213: (t4)V13L64O3Q3B8Q6B16Q4B&<C+&D&E&F+8>Q3B8A8Q6A16<Q4A&B&<C+&D
&E8>>Q3A8B8Q6B16Q4B&<C+&D&E&F+8>Q3B8A8Q6A16<Q4A&B&<C+&D&E8>>Q3A8
214: (t5)Q4L64Y51,15O2V14D8V13O3R16Q4{B&B&<C+&D&E&}16F+8R8O2V14D8
V13O4R16{A&A&B&<C+&D&}16E8R8O2V14D8V13O3R16Q4{B&B&<C+&D&E&}16F+8R
8O2V14D8V13O4R16{A&A&B&<C+&D&}16E8R8
215: (t6)V13F+V10F+F+F+V13EV10EEEV13F+V10F+F+F+V13EV10EEE
216: (t7)>V13BV10BBBV13<C+V10C+C+C+V13>BV10BBBV13<C+V10C+C+C+
217: (t8)V13DV10DDDV13DV10DDDV13DV10DDDV13DV10DDD
218: (t1)B<C+DC+>BAGF+GABGQ8A4T160RQ3V15<DT168
219: (t2)B<C+DC+>BAGF+GABGQ8A4R@39O4Q3V14D
220: (t3)B<C+DC+>BAGF+GABGQ8A4R@39O5Q3V15D
221: (t4)B8Q6B16>Q4B&<C+&D&E&F+8Q3B8B8Q6B16Q4G&A&B&<C+&D8>Q3B8G8Q
6G16<Q4E&F+&G&A&B8>Q3G8A8Q6A16<Q4A&B&<C+&D&E8>>Q3A8
222: (t5)O2V14D8V13O2R16{B&B&<C+&D&E&}16F+8R8O2V14D8V13O3R16{G&G&
A&B&<C+&}16D8R8O2V14D8V13O4R16{E&E&F+&G&A&}16B8R8O2V14D8V13O4R16{
A&A&B&<C+&D&}16E8R8
223: (t6)V13F+V10F+F+F+V13GV10GGGV13EV10EEEV13EV10EEE
224: (t7)V13>BV10BBBV13BV10BBBV13BV10BBBV13<C+V10C+C+C+
225: (t8)V13DV10DDDV13DV10DDDV13DV10DDDV13DV10DDD
226: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4T164RQ3DT168
227: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RQ3D
228: (t3)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+Q8
D4RQ3D
229: (t4)@95Q3L8O4V12AV9AAAV12AV9AAAV12AV9AAAV12AV9AAAV12AV9AAAV1
2AV9AAAV12AV9AAAV12AV9AAA
230: (t5)@95Q3L8O4V12F+V9F+F+F+V12DV9DDDV12GV9GGGV12F+V9F+F+F+V12
F+V9F+F+F+V12DV9DDDV12GV9GGGV12F+V9F+F+F+
231: (t6)@95O4V12DV9DDD>V12BV9BBB<V12C+V9C+C+C+V12DV9DDDV12DV9DDD
>V12BV9BBB<V12C+V9C+C+C+V12DV9DDD
232: (t7)@95O2V12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDDV12DV9
DDDV12DV9DDDV12DV9DDD
233: (t8)V14DV11DDDV14DV11DDDV14DV11DDDV14DV11DDDV14DV11DDDV14DV1
1DDDV14DV11DDDV14DV11DDD
234: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+L1
6D&C+&D&E&L8T140Q6DR
235: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+L1
6D&C+&D&E&L8Q6DR
236: (t3)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DDC+DEF+EDC+>B<C+DEC+L1
6D&C+&D&E&L8Q6DR
237: (t4)V12AV9AAAV12AV9AAAV12AV9AAAV12AV9AAAV12AV9AAAV12AV9AAAV1
2AV9AAA@40Q8V12<A4R4
238: (t5)V12F+V9F+F+F+V12DV9DDDV12GV9GGGV12F+V9F+F+F+V12F+V9F+F+F
+V12DV9DDDV12GV9GGG@40Q8V12<F+4R4
239: (t6)V12DV9DDD>V12BV9BBB<V12C+V9C+C+C+V12DV9DDDV12DV9DDD>V12B
V9BBB<V12C+V9C+C+C+@40Q8V12<D4R4
240: (t7)V12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDDV1
2DV9DDD@40Q8V12<D4R4
241: (t8)V14DV11DDDV14DV11DDDV14DV11DDDV14DV11DDDV14DV11DDDV14DV1
1DDDV14DV11DDDV14DRRR
242: /#3
243: (t1)T168@95L8O4V12Y48,6T160Q4B4F+4T168B4&Q7BR32.<C+32.E32&Q6
DR16C+16DR16>B16<F+4&Q5F+R16Q7D16
244: (t2)T168@96L8O4V12Y49,5Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
245: (t3)T168@95L8O4V12Y50,5Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
246: (t4)T168@96L8O4V12Y51,4Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
247: (t5)T168@95L8O3V11Y52,4Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
248: (t6)T168@96L8O3V11Y53,3Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
249: (t7)T168@95L8O3V11Y54,3Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
250: (t8)T168@96L8O3V11Y55,2Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16
DR16>B16<F+4&Q5F+R16Q7D16
251: (t1)T170Q4E4F+4Q3V13GV12F+EDQ4C+4F+4T168Q3EV12DC+D>
252: (t2)Q4E4F+4Q3V13GV12F+EDQ4C+4F+4Q3EV11DC+D>
253: (t3)Q4E4F+4Q3V13GV12F+EDQ4C+4F+4Q3EV12DC+D>
254: (t4)Q4E4F+4Q3V13GV12F+EDQ4C+4F+4Q3EV11DC+D>
255: (t5)Q4E4F+4Q3V12GV11F+EDQ4C+4F+4Q3EV11DC+D>
256: (t6)Q4E4F+4Q3V12GV11F+EDQ4C+4F+4Q3EV10DC+D>
257: (t7)Q4E4F+4Q3V12GV11F+EDQ4C+4F+4Q3EV11DC+D>
258: (t8)Q4E4F+4Q3V12GV11F+EDQ4C+4F+4Q3EV10DC+D>
259: (t1)T164Q4B4F+4T168B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+
4&Q5F+R16Q7D16
260: (t2)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
261: (t3)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
262: (t4)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
263: (t5)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
264: (t6)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
265: (t7)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
266: (t8)Q4B4F+4B4&Q7BR32.<C+32.E32&Q6DR16C+16DR16>B16<F+4&Q5F+R1
6Q7D16
267: (t1)T170Q4E4F+4Q3V13GV12F+EDT166Q4D4C+4T130Q2>B4&B@99O5L8Q3V
9Y48,5T168B
268: (t2)Q4E4F+4Q3V13GV12F+EDQ4D4C+4Q2>B4&BRY49,5
269: (t3)Q4E4F+4Q3V13GV12F+EDQ4D4C+4Q2>B4&BRY50,5
270: (t4)Q4E4F+4Q3V13GV12F+EDQ4D4C+4Q2>B4&BRY51,5
271: (t5)Q4E4F+4Q3V12GV11F+EDQ4D4C+4Q2>B4&BRY52,5
272: (t6)Q4E4F+4Q3V12GV11F+EDQ4D4C+4Q2>B4&BRY53,5
273: (t7)Q4E4F+4Q3V12GV11F+EDQ4D4C+4Q2>B4&BRY54,5
274: (t8)Q4E4F+4Q3V12GV11F+EDQ4D4C+4Q2>B4&BRY55,5
275: (t1)A+B<C+DC+>BA+G+A+B<C+>A+L16B&A+&B&<C+&>L8T164BBT168A+B<C
+DC+>BA+G+A+B<C+>A+L16B&A+&B&<C+&>T140L8BR16@95O5L8V12Y48,6T163Q6
C+16
276: (t2)R2R2R2R4.@1O5L8Q3V10Y49,8BA+B<C+DC+>BA+G+A+B<C+>A+L16B&A
+&B&<C+&>L8BR16@96O5L8V12Y49,5Q6C+16
277: (t3)@44O5L8Q3V9F+RRRC+RRRERRRDRRRF+RRRC+RRRERRRDRRR16@95O5L8
V12Y50,5Q6C+16
278: (t4)@44O4L8Q3V9BRRRBRRRA+RRRBRRRBRRRBRRRA+RRRBRRR16@96O5L8V1
2Y51,4Q6C+16
279: (t5)@45O4L8Q3V9DR>BR<ERC+RF+R>F+RBR<C+RDR>BR<ERC+RF+R>F+RBRR
R16@95O4L8V11Y52,4Q6C+16
280: (t6)R2R2R2R2R2R2R2R4.R16@96O4L8V11Y53,3Q6C+16
281: (t7)R2R2R2R2R2R2R2R4.R16@95O4L8V11Y54,3Q6C+16
282: (t8)@93O3L8Q4V10V10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DD
DV10DV7DDDV10DV7DDDV10DV7DDR16@96O4L8V11Y55,2Q6C+16
283: (t1)Q4C+4C+4Q8D4T168Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
284: (t2)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
285: (t3)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
286: (t4)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
287: (t5)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
288: (t6)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
289: (t7)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
290: (t8)Q4C+4C+4Q8D4Q7C+R16>B16<Q4C+4D4E4&Q5ER16Q7D16
291: (t1)T170Q4E4F+4>Q7B4<V11Q3C+DV12EV11DC+>BT168BA+T155Q6F+R16<
T168V13C+16
292: (t2)Q4E4F+4>Q7B4<Q3V11C+DV12EV11DC+>BBA+Q6F+R16<V13C+16
293: (t3)Q4E4F+4>Q7B4<Q3V11C+DV12EV11DC+>BBA+Q6F+R16V11A+16
294: (t4)Q4E4F+4>Q7B4<Q3V11C+DV12EV11DC+>BBA+Q6F+R16V11A+16
295: (t5)Q4E4F+4>Q7B4<Q3V10C+DV11EV10DC+>BBA+Q6F+R16V11<F+16
296: (t6)Q4E4F+4>Q7B4<Q3V10C+DV11EV10DC+>BBA+Q6F+R16V11<F+16
297: (t7)Q4E4F+4>Q7B4<Q3V10C+DV11EV10DC+>BBA+Q6F+R16V11<F+16
298: (t8)Q4E4F+4>Q7B4<Q3V10C+DV11EV10DC+>BBA+Q6F+R16V11<F+16
299: (t1)T166Q5C+4C+4T162Q8D4Q7T166C+R16>B16<T164Q5C+4T161D4T158E
4&Q5ER16Q7T163D16
300: (t2)Q5C+4C+4Q8D4Q7C+R16>B16<Q5C+4D4E4&Q5ER16Q7D16
301: (t3)Q5A+4A+4Q8B4Q7A+R16G+16Q5A+4B4Q8<C+2&
302: (t4)Q5A+4A+4Q8B4Q7A+R16G+16Q5A+4B4Q8<C+2
303: (t5)Q5F+4F+4F+2&Q5F+4F+4G2&
304: (t6)Q5F+4F+4Q8F+4Q7F+R16F+16Q5F+4F+4Q8G2
305: (t7)Q5F+4F+4Q8E+4Q7ER16D16Q5C+4>B4A+4&A4&
306: (t8)Q5F+4F+4Q8E+4Q7ER16D16Q5C+4>B4Q8A+4A4
307: (t1)Q6E4F+4T168Q4GQ3F+EDT160Q5D4T150C+4T130Q2>B4&B@99O5L8Q3V
9Y48,5T168B
308: (t2)Q6E4F+4Q4GQ3F+EDQ5D4C+4Q2>B4&BRY49,5
309: (t3)Q6C4C4Q3>B4&BBQ5B4A+4Q2F+4&F+RY50,5
310: (t4)Q6C4C4Q3>B4&BBQ5B4A+4Q2F+4&F+RY51,5
311: (t5)Q6G4&F+4D+4Q3EF+Q5G+4E4Q2D4&DRY52,5
312: (t6)Q8G4Q6F+4D+4Q3EF+Q5G+4E4Q2D4&DRY53,5
313: (t7)Q6D4&<D4Q3>G4&GF+Q5E+4F+4Q2B4&BRY54,5
314: (t8)Q8D4<D4Q3>G4&GF+Q5E+4F+4Q2B4&BRY55,5
315: /#4
316: (t1)@99O5L8Q3V9T168A+B<C+DC+>BA+G+A+B<C+>A+L16B&A+&B&<C+&>L8
T164BBT168A+B<C+DC+>BA+G+A+B<C+>A+L16B&A+&B&<C+&>L8T162BY48,8V10<
DT168
317: (t2)R2R2R2R4.@1O5L8Q3V10Y49,8BA+B<C+DC+>BA+G+A+B<C+>A+L16B&A
+&B&<C+&>L8BY49,5@98O5V8D
318: (t3)@44O5L8Q3V9F+RRRC+RRRERRRDRRRF+RRRC+RRRERRRDRRR
319: (t4)@44O4L8Q3V9BRRRBRRRA+RRRBRRRBRRRBRRRA+RRRBRRR
320: (t5)@44O4L8Q3V9R2R2R2R2R2R2R2R2
321: (t6)@45O4L8Q3V9DR>BR<ERC+RF+R>F+RBR<C+RDR>BR<ERC+RF+R>F+RBR<
DR
322: (t7)R2R2R2R2R2R2R2R2
323: (t8)@93O3L8Q4V10V10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDDV10DV7DD
DV10DV7DDDV10DV7DDDV10DV7DDD
324: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8T162DF+T168
325: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8D@101O5V8F+
326: (t3)V10ARRRERRRGRRRF+RRR
327: (t4)V10<DRRRDRRRC+RRRDRRR
328: (t5)V10ARRRBRRRARRRARRR
329: (t6)V10F+RDRGRERAR>AR<DR>DR
330: (t7)R2R2R2R2
331: (t8)V10DV7DDDV10DV7DDDV10DV7DDDV10DV7DDD
332: (t1)E+F+G+AV11G+F+E+D+V12E+F+G+E+
333: (t2)E+F+G+AV9G+F+E+D+V10E+F+G+E+
334: (t3)C+R>ARV10<DR>BRV11G+RBR
335: (t4)>F+RC+RV10DRG+RV11G+RG+R
336: (t5)@45V9>ARF+RV10BRG+RV11<C+R>C+R
337: (t6)Y53,0V9>ARF+RV10BRG+RV11<C+R>C+R
338: (t7)@100O4Q8L2V10F+&F+&V11E+
339: (t8)V9DV6DDDV10DV7DDDV11DV8DDD
340: (t1)V12L16F+&E+&F+&G+&L8F+AL16F+&E+&F+&G+&L8F+AF+G+V13AB<C+>
BV12AG+
341: (t2)V10L16F+&E+&F+&G+&L8F+AL16F+&E+&F+&G+&L8F+AF+G+V11AB<C+>
BV10AG+
342: (t3)AR@101Q3V8<C+C+C+C+C+C+C+C+C+C+C+C+C+C+
343: (t4)C+R@101Q3V8AAAAAAAAAAAABB
344: (t5)F+R@95V8Q8L4<F+&E+&E&D+&D&C+2
345: (t6)F+RR4R2R2R4V12C+R
346: (t7)V10F+&F+&F+E+
347: (t8)V11DV8DDDV11DV8DDDV11DV8DDDV11DV8DDD
348: (t1)V13L16F+&E+&F+&G+&L8F+AL16F+&E+&F+&G+&L8F+AF+G+V14AB<C+>
```

▶愛読者プレゼントにぜひコタツを載せて，この俺に暖かい冬を迎えるチャンスをくれー。
最近，とっても寒いんだよぉ。　　　　古川　泰寛　(19)　福岡県

```
BV13AG+
349: (t2)V11L16F+&E+&F+&G+&L8F+AL16F+&E+&F+&G+&L8F+AF+G+V12AB<C+>
BV11AG+
350: (t3)V9C+C+C+C+C+C+C+C+C+C+C+C+C+C+C+C+
351: (t4)V9AAAAAAAAAAAAAABB
352: (t5)Q3>F+8R8<V9F+&E+&E&D+&D&Q8C+2
353: (t6)F+RR4R2R2R4C+R
354: (t7)V11F+&F+&F+E+
355: (t8)V12DV9DDDV12DV9DDDV12DV9DDDV12DV9DDD
356: (t1)V14L16F+&E+&F+&G+&F+8A8F+&E+&F+&G+&F+8A8V14F+&E+&F+&G+&F
+8A8T165V15F+&E+&F+&G+&T162F+8A8
357: (t2)V11L16F+&E+&F+&G+&F+8A8F+&E+&F+&G+&F+8A8V12F+&E+&F+&G+&F
+8A8V13F+&E+&F+&G+&F+8A8
358: (t3)V10C+>AAAV11AAAA<V12CCCCV13C+C+DD
359: (t4)V10AF+F+F+V11F+F+F+F+V12AAAAV13AABB
360: (t5)Q3>F+8R8Q8V9F+&V10E+&E&V11D+&D&V12C+&>B
361: (t6)F+R@95V9Q8L4Y53,0F+&V10E+&E&V11D+&D&V12C+&>B
362: (t7)F+&V12F+&V13F+&V14F+4&V15F+4
363: (t8)V12DV8DDDV11DV9DDDV12DV10DDDV13DV11DDD
364: (t1)T158Y48,12Q8F+&G+&A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+F+&G
+&A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+
365: (t2)Y49,8Q8V14F+&G+&A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+F+&G+&
A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+
366: (t3)V14Q6C+Q8L16Y50,5>A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+F+&G
+&A&B<C+&>B&A&G+F+&G+&A&B<C+&>B&A&G+
367: (t4)V14Q6AR@97V14L8O4G+16&V13G+8.Q4V12ARQ8V14G+16&V13G+8.Q4V
12ARQ8V14G+16&V13G+8.Q4V12ARQ8V14G+16&V13G+8.
368: (t5)V14Q7L8F+R<F+4&F+RF+4&F+RF+4&F+RF+4&
369: (t6)V14Q7L8F+R<F+4&F+RF+4&F+RF+4&F+RF+4&
370: (t7)V14Q6F+8R8@97V14L8O4D16&V13D8.Q4V12C+RQ8V14D16&V13D8.Q4V
12C+RQ8V14D16&V13D8.Q4V12C+RQ8V14D16&V13D8.
371: (t8)V14DR@97V14L8O3B16&V13B8.Q4V12ARQ8V14B16&V13B8.Q4V12ARQ8
V14B16&V13B8.Q4V12ARQ8V14B16&V13B8.
372: (t1)V14F+&G+&A&B<T154V15C+&D&E&DT158C+&>B&A&GV14T154F+&E&T15
0D&C+
373: (t2)F+&G+&A&B<V15C+&D&E&DC+&>B&A&GV14F+&E&D&C+
374: (t3)F+&G+&A&B<V15C+&D&E&DC+&>B&A&GV14F+&E&D&C+
375: (t4)Q4V13ARV15G8&V13G8&Q6G4Q8V14G16&V13G8.
376: (t5)F+RV15>A4&A4V14A4
377: (t6)F+RV15>A4&A4V14A4
378: (t7)V13C+RV15E8&V13E8&Q6E4Q8V14E16&V13E8.
379: (t8)V13ARV15<C+8&V13C+8&Q6C+4Q8V14C+16&V13C+8.
380: /#5
381: (t1)T168Q8L8D4R@101V14O6Q3Y48,15T160DT168
382: (t2)Q8L8D4R@101V14O5Q3Y49,5D
383: (t3)Q8L8D4R4
384: (t4)V12F+4R4
385: (t5)@101L8O2Q3V14AV12AAA
386: (t6)@101L8O2Q3V14DV12DDD
387: (t7)@101L8O2Q3V15DV13DDD
388: (t8)@93O3L8Q4V15DV13DDD
389: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
390: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
391: (t3)@97V13Q6L8O4Y50,0D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&
AR16F+16
392: (t4)@97V13Q6L8O4Y51,15D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4
&AR16F+16
393: (t5)@95O3V13L2Q7AAAA
394: (t6)@95O3V13L2Q7DDDD
395: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
396: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
397: (t1)C+DEF+EDC+>B<C+DEC+Q8D4T164RQ3DT168
398: (t2)C+DEF+EDC+>B<C+DEC+Q8D4RQ3D
399: (t3)Q6G4A4Q5BQ4AGF+Q6E4A4Q4GF+EF+
400: (t4)Q6G4A4Q5BQ4AGF+Q6E4A4Q4GF+EF+
401: (t5)AAAA
402: (t6)DDDD
403: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
404: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
405: (t1)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
406: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
407: (t3)Q6D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
408: (t4)Q6D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
409: (t5)AAAA
410: (t6)DDDD
411: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
412: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
413: (t1)C+DEF+EDC+>B<C+DEC+Q8D4T162RQ3>AT168
414: (t2)C+DEF+EDC+>B<C+DEC+Q8D4RQ3>A
415: (t3)Q6G4A4Q5BQ4AGF+Q6F+4E4Q7D4&DR16Q6E16
416: (t4)Q6G4A4Q5BQ4AGF+Q6F+4E4Q7D4&DR16Q6E16
417: (t5)AAAA
418: (t6)DDDD
419: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
420: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
421: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
422: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
423: (t3)Q6E4E4Q7F+4Q6ER16D16Q6E4F+4Q7G4&GR16Q6F+16
424: (t4)Q6E4E4Q7F+4Q6ER16D16Q6E4F+4Q7G4&GR16Q6F+16
425: (t5)AAAA
426: (t6)DDDD
427: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
428: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
429: (t1)B<C+DC+>BAGF+GABGL16A&G&A&B&L8T164AAT168
430: (t2)B<C+DC+>BAGF+GABGL16A&G&A&B&L8AA
431: (t3)Q6G4A4Q7D4Q4EF+Q5GQ4F+EDDC+>Q7AR16Q6<E16
432: (t4)Q6G4A4Q7D4Q4EF+Q5GQ4F+EDDC+>Q7AR16Q6<E16
433: (t5)AAAA
434: (t6)DDDD
435: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
436: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
437: (t1)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
438: (t2)Q6B&B16.<C+32&>Q7B4Q8A4RQ3AQ6B&B16.<C+32&>Q7B4Q8A4RQ3A
439: (t3)Q6E4E4Q7F+4Q6ER16D16Q6E4F+4Q7G4&GR16Q6F+16
440: (t4)Q6E4E4Q7F+4Q6ER16D16Q6E4F+4Q7G4&GR16Q6F+16
441: (t5)AAAA
442: (t6)DDDD
443: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
444: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
445: (t1)B<C+DC+>BAGF+GABGQ8A4T165R<Q3V15D
446: (t2)B<C+DC+>BAGF+GABGQ8A4R<Q3V15D
447: (t3)Q6G4A4Q5BQ4AGF+Q6F+4E4Q7D4&DR
448: (t4)Q6G4A4Q5BQ4AGF+Q6F+4E4Q7D4&DR
449: (t5)AAAA
450: (t6)DDDD
451: (t7)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
452: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
453: (t1)T169C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
454: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
455: (t3)V14D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
456: (t4)V14D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
457: (t5)V14AAAA
458: (t6)V14DDDD
459: (t7)V15>D<V13DDDV15>D<V13DDDV15>D<V13DDDV15>D<V13DDD
460: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
461: (t1)T170C+DEF+EDC+>B<C+DEC+Q8D4T167RQ3D
462: (t2)C+DEF+EDC+>B<C+DEC+Q8D4RQ3D
463: (t3)Q6G4A4Q5BQ4AGF+Q6E4A4Q4GF+EF+
464: (t4)Q6G4A4Q5BQ4AGF+Q6E4A4Q4GF+EF+
465: (t5)AAAA
466: (t6)DDDD
467: (t7)V15>D<V13DDDV15>D<V13DDDV15>D<V13DDDV15>D<V13DDD
468: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
469: (t1)T171C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
470: (t2)C+DEF+EDC+>B<C+DEC+L16D&C+&D&E&L8DD
471: (t3)Q6D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
472: (t4)Q6D4>A4<Q7D4&DR16Q6E16F+R16E16F+R16D16A4&AR16F+16
473: (t5)AAAA
474: (t6)DDDD
475: (t7)V15>D<V13DDDV15>D<V13DDDV15>D<V13DDDV15>D<V13DDD
476: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
477: (t1)T173C+DEF+EDC+>B<C+DT169EC+
478: (t2)C+DEF+EDC+>B<C+DEC+
479: (t3)Q6G4A4Q5BQ4AGF+Q6F+4E4
480: (t4)Q6G4A4Q5BQ4AGF+Q6F+4E4
481: (t5)AAA
482: (t6)DDD
483: (t7)V15>D<V13DDDV15>D<V13DDDV15>D<V13DDD
484: (t8)V15DV13DDDV15DV13DDDV15DV13DDD
485: (t1)T174L16V15D&C+&D&E&D8F+8D&C+&D&E&L8DF+T175DEF+GQ4AQ3GF+E
486: (t2)Y49,10V15L16D&C+&D&E&D8F+8D&C+&D&E&L8DF+DEF+GQ4AQ3GF+E
487: (t3)@101V14O4Q3Y50,5L16D&C+&D&E&D8F+8D&C+&D&E&L8DF+DEF+GQ4AQ
3GF+E
488: (t4)@97L8Q4O4Y51,5V13DV11DDDV13DV11DDDV13DV11DDV12DV15C+V12C
+C+V11C+
489: (t5)@95L2Q7O3V14AAAA
490: (t6)@101L8Q4O3V13F+V11F+F+F+V13F+V11F+F+F+V13F+V11F+F+V12F+V
15GV12GGV11G
491: (t7)@101L8Q5O2V14DDDDC+C+C+C+>BBBBV15AAV14AA
492: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
493: (t1)T177L16D&C+&D&E&D8F+8D&C+&D&E&L8DF+T178DEF+GQ4AQ3GF+E
494: (t2)L16D&C+&D&E&D8F+8D&C+&D&E&L8DF+DEF+GQ4AQ3GF+E
495: (t3)L16D&C+&D&E&D8F+8D&C+&D&E&L8DF+DEF+GQ4AQ3GF+E
496: (t4)V13DV11DDDV13DV11DDDV13DV11DDV12DV15C+V12C+C+V11C+
497: (t5)AAAA
498: (t6)V13F+V11F+F+F+V13F+V11F+F+F+V13F+V11F+F+V12F+V15GV12GGV1
1G
499: (t7)V14<DDDDC+C+C+C+>BBBBV15AAV14AA
500: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
501: (t1)T180L16D&C+&D&E&L8T179DBT181AGF+ET182L16D&C+&D&E&L8T181D
F+T183EDC+>B
502: (t2)L16D&C+&D&E&L8DBAGF+EL16D&C+&D&E&L8DF+EDC+>B
503: (t3)L16D&C+&D&E&L8DBAGF+EL16D&C+&D&E&L8DF+EDC+>B
504: (t4)V15DV13DDDV14DV13DDDV15DV13DDDV14DV13DDD
505: (t5)@101L8Q4O3V15AV13AAAV14AV13AAAV15AV13AAAV14AV13AAA
506: (t6)V15F+V13F+F+F+V14F+V13F+F+F+V15F+V13F+F+F+V14F+V13F+F+F+
507: (t7)Q8V15DQ7V14DQ6DQ5DV15DV14DDDQ8V15DQ7V14DQ6DQ5DV15DV14DDD
508: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
509: (t1)T184L16A&G&A&B&L8T183A<DT185C+>BAGT186L16F+&E&F+&G&L8T18
5F+BT187AGF+E
510: (t2)L16A&G&A&B&L8A<DC+>BAGL16F+&E&F+&G&L8F+BAGF+E
511: (t3)L16A&G&A&B&L8A<DC+>BAGL16F+&E&F+&G&L8F+BAGF+E
512: (t4)V15DV13DDDV14DV13DDDV15DV13DDDV14DV13DDD
513: (t5)V15AV13AAAV14AV13AAAV15AV13AAAV14AV13AAA
514: (t6)V15F+V13F+F+F+V14F+V13F+F+F+V15F+V13F+F+F+V14F+V13F+F+F+
515: (t7)Q8V15DQ7V14DQ6DQ5DV15DV14DDDQ8V15DQ7V14DQ6DQ5DV15DV14DDD
516: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
517: (t1)T182DV12L16Y48,10T188Q7C+C+DDC+C+DQ4DQ7V13EEF+F+EEF+Q5F+
Q7V14AAAAAAAQ6AQ7<V15C+C+DDC+C+
518: (t2)DV11L16Y49,5Q7AAAAAAAQ4AQ7V12<C+C+DDC+C+DQ5DQ7V13EEF+F+E
EF+Q6F+Q7V14AAAAAA
519: (t3)DV11L16Y50,5Q7<EEF+F+EEF+Q5F+Q7V12AAAAAAAQ6AQ7V13<C+C+DD
C+C+DDV14EEF+F+EE
520: (t4)D@101L16Q7O4V11C+C+DDC+C+DDV12EEF+F+EEF+F+V13EEF+F+EEF+F
+V14AAAAAA
521: (t5)AL16Q7V11AAAAAAAAV12AAAAAAAAV13AAAAAAAAV14AA<DD>AA<
522: (t6)F+V11L16Q7>AA<DD>AA<DDV12>AA<DD>AA<DDV13>AA<DD>AA<DDV14>
AA<DD>AA<
523: (t7)Q6D>V12A<D>A<D>V13A<D>A<D>V14A<D>A<D>V15A<D>A<
524: (t8)V15DV13DDDV15DV13DDDV15DV13DDDV15DV13DDD
525: (t1)@95Y48,10L8Q4T184DRERDRERQ6T178F+RRRT160ARRRD4RR
526: (t2)@95Y49,7L8Q3ARARF+RARQ6ARRR<C+RRR>F+4RR
527: (t3)@95L8Q4F+RV15ERDRERQ6F+RRRARRRD4RR
528: (t4)@95L8Q3ARC+R>BR<C+RQ6C+RRRERRR>A4RR
529: (t5)@95L8Q3DRARF+RARQ6ARRRERRRF+4RR
530: (t6)@95L8Q4DR<ERDRERQ6F+RRR>ARRR<D4RR
531: (t7)@95L8Q4DR<ARBRARQ6F+RRR>ARRR<D4RR
532: (t8)DR@95L8Q4O3V15ARBRARQ6F+RRR>ARRR<D4RR
533: (p)
```

特集

いきなり初春(はる)からハードウェア

パソコンなくして人生はありえないという皆さんでも，ハードウェアに詳しい人は少ないでしょう。大事なのはソフトウェアで，だからパソコン雑誌もソフトの話が中心，というのが一般的な風潮のようです。しかし，自分だけのパソコンを持ち，愛機の機能を引き出したいと考えているOh!X読者の皆さんは，ハードとソフトの境目にいるとはいえないでしょうか。ハードウェアには，複雑で難しいものというイメージがあるかもしれません。また，機械的なもの，電気的なものに対する苦手意識が，理解を妨げている場合もあるでしょう。それでも，ひとたび，ハードウェアの世界に足を踏み入れてデジタル回路を覗いてみれば，そこにはすでに見なれたソフトウェアが存在することに気づくはずです。そしてもちろん，ハードを触ることは純粋に楽しいことでもあるのです。ハードウェアを理解する心があれば，いままで以上にパソコンを楽しむことができるでしょう。コンピュータはソフトとハードとでできているのですから。

CONTENTS

●デジタル回路入門編

斎場パンクローとジョセフソン・素子の？？？

ハードウェアをめぐる冒険

Ogikubo Kei
荻窪　圭

ソフトしかわからない少年と，ハードしかわからない純粋な少女。だが，わずか5Vの電気の流れの中にも不思議なソフトウェアが存在した。果たして，ハードウェアと呼ばれるものの正体はなにか。荻窪圭氏が論理回路の世界に挑む，謎の散文。

忘れ去られたハードウェア

遠い昔，ハードを知ることからすべてのマイコンは入門された。『初歩のラジオ』が今の『I/O』くらい分厚くて，電波新聞社の雑誌といえば『マイコン』ではなく『ラジオの製作』であった時代だ。

さらにずっと遠い昔，世界最初の計算機ENIACはフォン・ノイマン型ではなかった。つまりストアードプログラム方式（プログラムを入力して溜めておき，それを何度も実行するという方式）ではなかったのである。「世界最初の非ノイマン型コンピュータは？」，「ENIAC」という古典的ギャグもあるくらいだ。

ではどうやっていたかというと，もはや第2次大戦直後の話で，そのころ私の親父さんは焼夷弾から逃げのびた小学生だったからそんな昔のことなどわからないが，きっと，配線をこちょこちょやったり，スイッチをパチパチやったりしてアルゴリズムを作って計算させていたのだろう。うーん，気が遠くなる話。

でも，あまり昔でない昔，シンセサイザといえば，アップライトピアノよりもでかくて，パネルにはたくさんのジャックとしっぽのついたプラグがあって，どれをどこに差し込むかという，昔の映画に出てくる電話の交換手みたいな作業をして配線を変え，音色を作り出していた。あのケーブルがうねうねする姿は目に焼きついて忘れられない。現在のようにROMカートリッジに音色が何百，5インチディスクに音色が何千というのはノストラダムスだって思いもしなかったろう。

しかし，文明というのは，文明堂のカステラがいつまでも変わらないのとは正反対に，トットコトットコと資本主義の法則にのっとって先走りしてしまうものだ。ハードのハの字も知らなくとも，家庭用電源100Vは実は片方がアースになっていることを知らなくとも，パソコンを使うとなぜFM放送にノイズが増えるかも知らなくたって，構わない時代になってしまったのである。

そう，君は，松ヤニの焦げる匂いに恍惚となったことがあるか。

巷におけるハードとソフト

巷では，最近，ソフトがない！　という危機感が漂っているらしい。とはいっても，パソコンの世界のことではない（パソコン界では，いいソフトがないのは結構当たり前のことだとみんな思っている)。日本各地で地方イベントが開催されても，催しものはみんな似たりよったりで面白味がない。各地に立派な劇場やコンサートホールが次次と建つのに，上演する（客を呼べるだけの）劇団や（新しい）ミュージシャンがいない。FM放送局がどんどん開局しても，流すべきいい音楽がない。といったことが今ごろになってやっと問題視されてきているのだ。

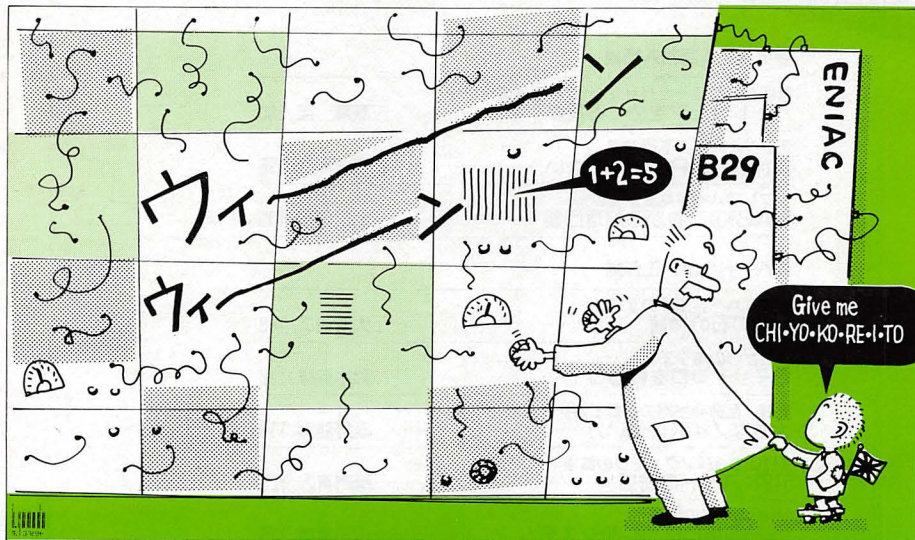
こういった問題は，入れ物があっても中に入れるものがないという意味でパソコンと一緒である。パソコン界は進んでいたのだ，時代の最先端だったのだ，トレンドなのだ，と思ってよろしい。

で，パソコン界を見てきた経験からいって，きっと，いいソフトは文化鎖国国家日本ではなかなか生まれないでしょう。生まれても，いいもの，面白いものを見分ける目と暇が日本の中流市民にはないからね。いくらFM局がいっぱいあっても，同じようなものばかりがかかっているようでは聴く気をなくすよ。

メニコンO₂というハード

コンタクトレンズといえば，ハードとソフトの2種類である。どちらも見た目は一緒だが，触ると違う。ソフトは柔らかいがハードは硬い。

たいていの人はコンタクトレンズを落とすと，見ず知らずの他人まで巻き込んでまで大騒ぎする。朝のラッシュの満員JRでコンタクトを落として大騒ぎし，何人ものサラリーマンを遅刻させた女子高生もいた。目の中に入れても痛くないとは，きっとコンタクトのことなのだろう。そして実は落としやすいのはハードのほうなのである。ソフトは人間の目に馴染みやすいが，ハードは結構わがままで，はずみで簡単にはずれてしまう。酸素を通すメニコンO_2とてハードには変わりないから落としやすい。

ちなみに，ハードコアとよくいうが（もう死語かな)，あれは訳すと，ハードの核，つまりCPUのことであるからよく覚えておくように（編集部：ウソつき!!)。

本来、ハードとソフトは対等のはず

コンピュータの話をしよう。ハードとソフトはどっかのメーカーの戦略かどっかの雑誌の陰謀か言葉の持つ運命か知らないけれど，ことごとく対比されてきた。そして，行司はソフトに軍配を上げ，物言いはつかなかった。なぜなら，ソフトは圧倒的にアメリカに遅れていたから。だが,「ソフトがない！」的危機感が広まったからといって，いいソフトがたくさんできたかというとそうでもないわけで，一太郎をありがたがっているんだから，しょせんはその程度の危機感だったのである。合掌。

また話がそれてしまった。申し訳ない。今回はハードウェアの特集なんだから，さっさとハードの話でもしましょう。まずは常識から。

その１：ハードウェアもソフトがなければただの箱である。

その２：ソフトウェアはハードなしには存在しえない。

仮想マシンのソフトはハードがなくても存在するぞ，などとはいわないように。ソフトの入ってないハードも実在しないのだから。

さて、デジタルとアナログ

この世はデジタル時代である。たとえば時間だ。一時期流行ったデジタル時計がダサいということになって再びアナログが主流だが，とはいえ，「今，何時？」と聞かれて「６時53分」なんて答えているようではデジタルの世界に首まで漬かっているといってよい。「そうね，だいたいねー」とか「まーだ，はやいー」と答えるのが正しいアナログである。

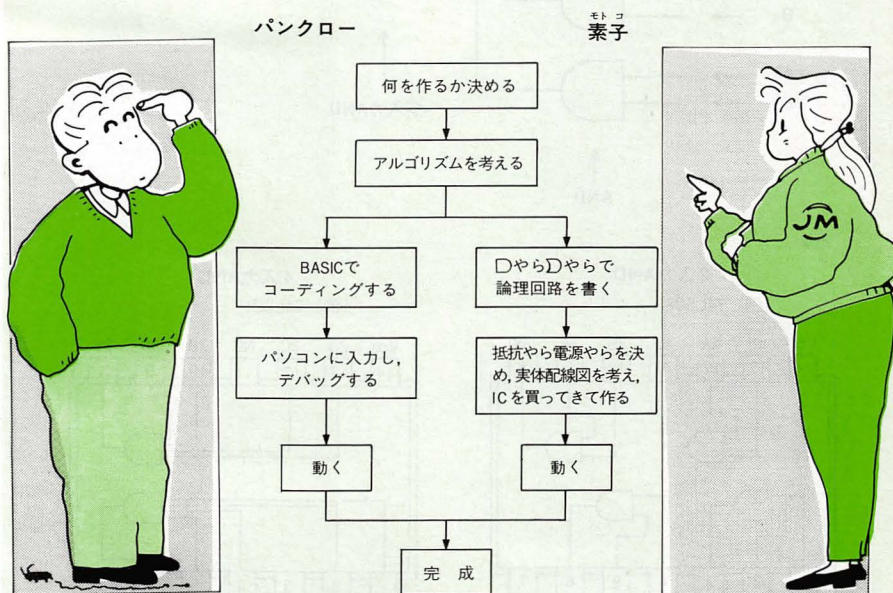
実際のところ，人にはアナログのほうが気持ちいいのだが，デジタルが細かくなりすぎて人の感覚器官にはそれがデジタルかアナログか区別できなくなっているくらい文明は進んでいる。CDなどはデジタルであるから，滑らかでなくデコボコなはずなのだが，あまりにデコボコが細かすぎるから人の耳には滑らかに聞こえてしまう，といったようなものだ。で，デジタルにすると，何が都合がいいかというと，デジタルだと情報量がコンパクトに収まって，加工しやすくて，いくつでもコピーが取れるなどなどだ。

言い換えよう。時代はアナログからアナログのように見えるデジタルへという流れにすでに乗っかってしまっているのだ。

ところが，逆もまた真である。世の中の森羅万象はアナログである。文明の誇るデジタルというのはもともとアナログなものをデジタルとみなして処理し，人々に見せるときには再びアナログっぽくしているのだ。

だいたい考えてみればいい。電線を流れる電気は１か０なんてことはありえないではないか。電池だって1.5Vなどといっておきながら使っているうちに1.3Vになり，懐中電灯の光は弱くなる。

で，デジタル回路もまた同じである。アナログなものをムリヤリ１と０にしちまっているのだ。０は0V。１は5Vとハッキリしていればまだデジタルっぽい。が，アナログであるから，正確に０とか５であるわけがない。正しくは低い電圧は０，高い電圧（といってもあまり高いと回路が壊れる）は１として処理してしまおうというのである。アナログに電流を処理するトランジスタをデジタルに使ってしまっているのだ。おお，文明よ。

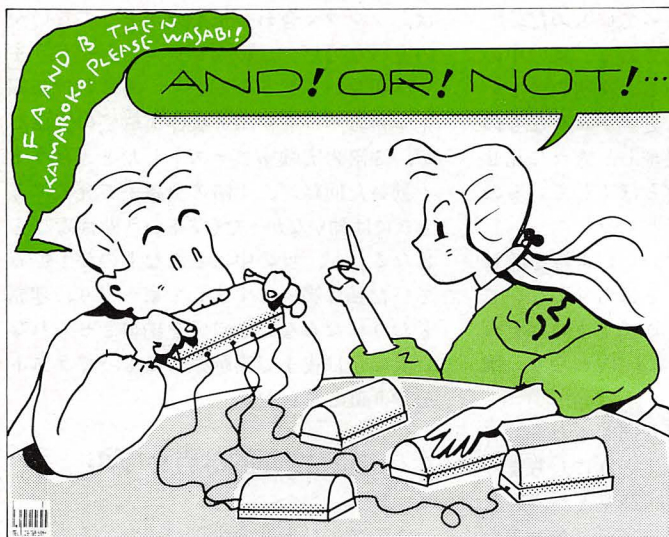
だから，電気の流れる回路といっても，設計側は１と０の組み合わせだけを考えればよい。アナログな電気の話はあとでツジツマを合わせればなんとかなるのである（たいていの面倒臭い電気的問題は，こんなときはこんな抵抗でこんなコンデンサを入れて，こうすれば大丈夫，といったパターンで切り抜けられる）。

言ってしまおう。ソフトとハードの違いは，ツジツマ合わせが必要かそうでないかの違いなのだ。このツジツマ合わせがうまくいかないとどうなるか？　ある数理情報工学科のハードウェア製作実験での実話だが，５階の実験室でテストしたときはうまく動いた回路が，１階の会議室で発表するときには動かなかったなどという悲惨なことになる。が，世の中はそんなものだと悟っていた私は笑ってすまして家へ帰り，理屈どおりにならないと自分を納得させられない女の子は夜まで実験室にこもってテストをやり直していた。

ある古典的なカップル

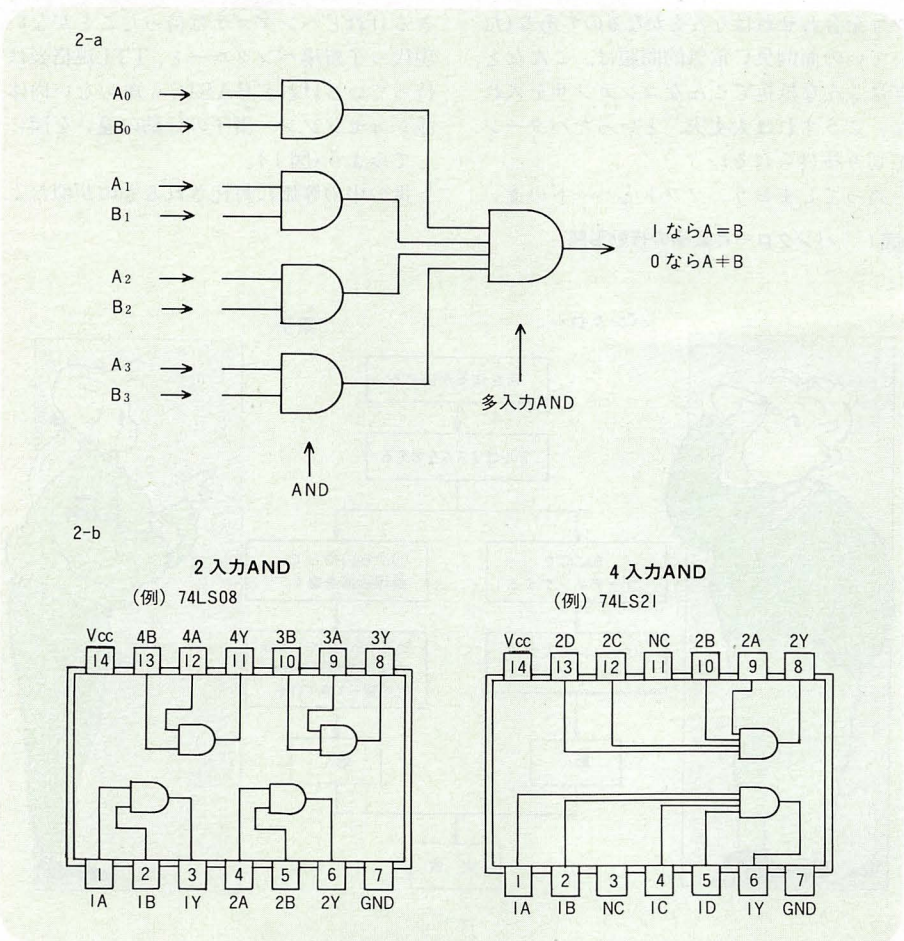
ここに，ひとりの少年がいた。その名を斎場（サイバー）パンクローという。彼にはなんと，恋人がいた。日本の女子高生みたいにパソコン少年なんて暗くてダサいなどとのたまわない日系２世のまっとうな女の子。その名もジョセフソン・素子（モトコ）（昔，SFファンの間で流行った古典的ギャグである。これを最初にいったのは誰か，知っている方は教えてください）である。

さて，パンクロー少年はどうなったか。その彼女は何をしているのか。BASICはできるけれどハンダゴテは持ったことがない現代っ子斎場パンクローと，TTL規格表は持っているけれどBASICも知らない肉体派ジョセフソン・素子（モトコ）の行動の違いを図にしてみよう（図１）。

世の中の等価に対比されるものが似たよ

図１　パンクローと素子の行動形態

うなパターンを示すのと同じように，彼らの行動もまたそっくりである。

素子はハードウェア恐怖症のパンクローにいった。

「あなたがこれこれこれがこうなるとあれがああなるから，ここでこれが計算できて……と考えて楽しいのと，私がここでこうするにはこの素子とあの素子を組み合わせてこうすればここにはこれが出力されるわ，と考えて楽しいのと一緒なのよ」

「でも，なんか面倒臭そう」と，パンクロー。彼は，いつ，電流増幅率とかエミッタフォロワー回路だとかパスコンだとかいうわけのわかんない用語が出てくるかと気が気でないのだ。それに，なにか考えるたびにICの規格表とにらめっこなんて面倒だと信じていた。

「だって，結局はゴキブリのお化けみたいなのをいっぱいくっつけて，動けないようにハンダづけして，蓋を開けたゴキブリホイホイみたいなのを作るんだろ」

素子はそれを聞いてむっとした。私が作っているのは台所の隅に転がっているゴキブリホイホイだというのね。

「このー，マルチバイブレーター光線！」

「ぎゃあああああ」

図2　ANDのカマボコの機能

あんどとおあとのっととそれから…

デジタル回路，というか論理回路には2つある。組み合わせ回路と順序回路である。組み合わせのほうは実に単純である。皆さんお馴染みのアンドやらオアやらの組み合わせで回路を作るのである。BASICでいう，

IF A AND B THEN……

というのは，AとB共に真であったときのみTHEN以下が実行されるということであるから，AもBも1ビットのデータだったとすると，判断部分はカマボコ1個である。AもBも4ビットのデータだったとすると，たとえば，カマボコは5個でできる。図2-aをご覧あれ。これを実現しようとすると，ICは2つ必要である。なんと，ANDが4つ入っているICやら4入力ANDなるICがあるからである。

4入力AND（74LS21）の片方が余るではないか，もったいない。と，思う必要はない。半導体は安いのであるから，湯水のように使うのもよろしい。さらには8入力NANDという強の者もいるので，その先にNOTひとつをつけただけで同じものを作れるのがデジタル回路の恐ろしいところだ。

「組み合わせ論理回路の文法はBASICなんかよりずっと簡単明瞭なのよ」

と，素子はいった。

「ANDでしょ，ORでしょ，それからXOR。それぞれのNOT版に独立したNOTのインバータの7つだけ」

一応、数学の話です

ブールさんという変な名前の人がいて，彼の名がとられたブール代数という理論がこの世に存在する。2進数を使った論理演算の理論である。ブールさんより変な名前のド・モルガンという有名な人もいて，ド・モルガンの定理なるものもある。

こういったものは，小学校か中学校で習

図3　ゲートの真理値素

A B	AND	OR	XOR	NAND	NOR
0 0	0	0	0	1	1
0 1	0	1	1	1	0
1 0	0	1	1	1	0
1 1	1	1	0	0	0

・NANDはANDのNOT版
・NORはORのNOT版

った集合の親戚だと思えば話は早い。ANDは論理積だから“・”，ORは論理和だから“＋”である。1×1は1だけど，1に0を掛けたら0であるからANDは論理積なのである。同様に，0に1を足したら1だからORは論理和なのである。

また，XORは排他的論理和という。これは図3の真理値表を見てもらえばわかるとおり，入力の2つが同じなら出力は0だよん，という他人と違っていればよしとするわがままな関数である。これは記号がなくて困ったのかどうかしらないが，「⊕」と表す。ちなみに，XORのNOT版は入力が一致していればとにかく出力は1になるという，日本的関数である。で，NOTは上にバー「￣」をつけることで表せる。ひとつならNOT，2つつければ元のままで，3つつければまたNOT，4つもつけると汚くて読みにくい。論理式というやつはこういった記号で表される。どれも1か0の状態しか持たないから，非常に単純である。図2-aを論理式にすると，

$(A_0 \cdot B_0) \cdot (A_1 \cdot B_1) \cdot (A_2 \cdot B_2) \cdot (A_3 \cdot B_3)$＝出力

となる。

真理値表というやつは，つまり，アドベンチャーゲームを作るときなんか便利である。石とお金と棒を持っているか，秘密の鍵を持っているか，うんちゃらとうんちゃらをしているときは扉の向こうへ行ける，なんてときは多入力の真理値表を書いて，そこから必要な論理を導き出すのだ。入力が多いと論理式は複雑になるが，全部を並べてIF文にするのは愚の骨頂である。なんとか簡単な論理にして，IF文は少なくしたいものだ。

それにはカルノーマップやらクワイン・マクラスキの簡単化法などという方法なんかが知られているので，これは，パズルみたいで面白いから，興味ある人は調べてみるといいだろう。フラグだらけの極悪なプログラムを組む人には，必需品かもしれない。

図4　扉が開く条件

論理式

扉が開く＝守護霊・鏡・剣・玉＋$\overline{\text{守護霊}}$・剣・玉＋守護霊・$\overline{\text{鏡}}$・$\overline{\text{剣}}$＋鏡・剣・$\overline{\text{玉}}$＋鏡・$\overline{\text{剣}}$・玉

BASIC風に書くと

```
if  (守護霊and鏡and剣and玉) or
    (not(守護霊)and剣and玉) or
    (守護霊and not(鏡)not(剣))or
    (鏡and剣and not(玉))or
    (鏡and not(剣)and玉)
  then  扉が開く
```

論理回路で「開け、ごま！」

ここで突如，クリアしなければならないイベントが登場したとしよう。クリアできる条件は，4つのアイテムの組み合わせのうちの6通りである。アイテムは，守護霊，鏡，剣，玉となっている。

4つのアイテムを全部持っている場合か，守護霊がなくて剣と玉を持っている場合か，守護霊を持っていて鏡と剣がない場合か，鏡と剣を持っていて玉がない場合か，鏡と玉を持っていて剣がない場合のいずれかを満たしたときに天国の扉は開かれる(図4)。

これは，結局，

扉が開く＝NOT守護霊＆剣＆玉＋守護霊＆NOT鏡＆NOT剣＋鏡＆剣＋鏡＆玉

＝$\overline{\text{守護霊}}$・剣・玉＋守護霊・$\overline{\text{鏡}}$・$\overline{\text{剣}}$＋鏡・(剣＋玉)

になるのである。

このように，デジタル回路の基礎である組み合わせ論理回路はIF文の塊であったわけだ。

これを論理回路にすると図5になる。BASICで書くと，図6になる。aは式をそのままIF文にしたものだが，bは論理回路のほうをフラグを使って表したものである。こうしたほうが作る側は混乱しなくてよい。

世の中はそういったものである。で，論理回路で表せるものは，その働きをするICを買ってくれば作れるのである。作る話はきっと，後ろのページで誰かがやってくれているだろう。

基本は足し算である

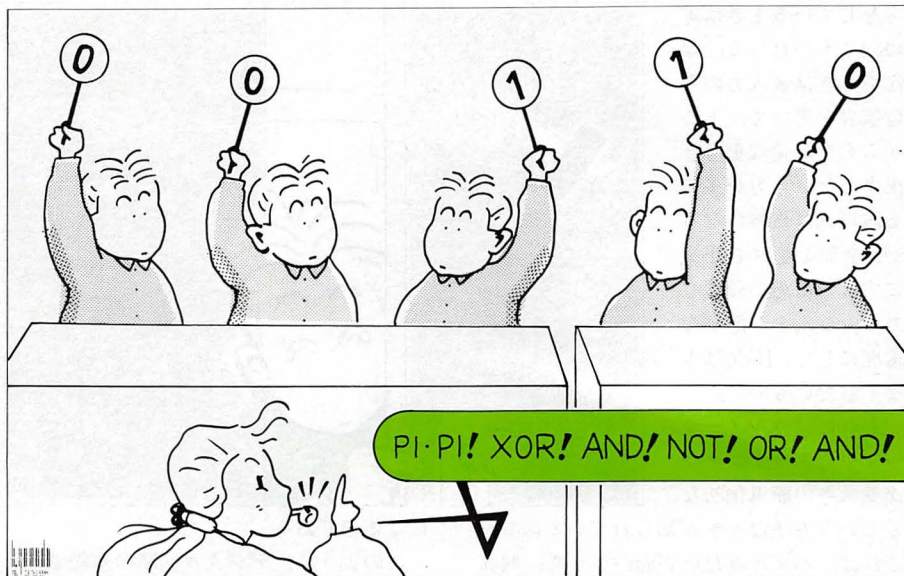
さて，1000011＋10011は，もちろん1010022である。というのは冗談で，1010110であることは2進数の世界では常識である。

Z80なんかだと，ADD A, Bなどとやって，あらかじめレジスタに値をセットしておくのだが，こいつをデジタル回路にしてしまうことは簡単で，頭の体操にもなって，のち

図5　組み合わせの論理回路

図6　組み合わせのIF文

```
6-a
if  (not(守護霊)and剣and玉) or
    (守護霊and not(鏡)and not(剣))
    (鏡and(剣or玉)) or
  then  扉が開く

6-b
if  not(守護霊)and剣and玉   then フラグ1=-1
if  守護霊and not(鏡)and not(剣)  then フラグ2=1
if  鏡and(剣or玉)  then フラグ3=1
if  フラグ1 or フラグ2 or フラグ3  then 扉が開く
```

のち便利なのである。

で，ANDやらORさんたちは一度に1ビットしか扱えないから，1ビットの足し算を作って，それをたくさんくっつけてやればいいことは自明である。これはBASICなんかで10進数で1000桁の足し算をやりたい！ などと思い立ったときのことを考えてもらえばよい。

では，斎場パンクロー君はどうしますか。

「まず，大きさが1000の1次元配列を3つDIMして，ひとつを1桁と考えて，下の桁，つまり添え字0からFOR～NEXTで1000回ループさせながら各桁ごとに計算していくんだけれど，結果が10を超えたら，どっかの変数に超えた分だけ入れておいて，次の桁のとき忘れないように一緒に足してやればできると思います」

「そんなの，あったりまえよね」

といったのは出番のないジョセフソン・素子。

確かにそのとおりである。論理回路で，つまりデジタル回路で2進の足し算をしようなどと思い立った場合もまったく同じことをしてやればいい。

入力はその桁のA，Bと，下からの繰り上がりの3つ，出力はその桁の結果と，繰り上がりの2つである。

とりあえず，BASICのように勘で適当にヤリながら試行錯誤でうまくいく方法を探すなどという怠慢はハードウェアの場合許されないからして，きちんと，そこいらへんに転がっている広告の裏でもくしゃくしゃになったリストの裏でもOh!Xの隅の余白にでも手のひらでもいいから，真理値表を書いてみる(図7)。

下からの繰り上がりがないとき，和はXORであることがすぐにわかる。AとBが違うときは結果が1になるのである。で，AもBも1のとき，繰り上がりを1にしてやればいい（AND)。逆に，下からの繰り上がりがあったら結果はXORの逆(NOT)になり，繰り上がりはAとBのORである。

足した結果をまず出すことにする。とりあえず，AとBのXORをとり，その結果と下からの桁上がりのXORをとればいいのは誰にでも……パンクロー，わかるか？

「わかんねえよ！ んな面倒臭えこと。脳がバイオソフトに犯されたみてえだ」

では，下からの繰り上がりとAとBのXORの真理値表を書いてみよう。

ほら，なった。

「不思議だが本当だ」

と，パンクロー。

で，和を求める部分は図8-aになる。

で，繰り上がりはさっきのようにはいかない。同じ手を使おうとすると，こける。わかるか，パンクロー。

「わからねえ」

さては，おまえ，頭使ってないな。のんべんだらりと聞いておるだろ。

で，こんなやつはほっといて，簡単にいうと，AとBの結果と下からの繰り上がりの2つからは一意に求まらないからである。

で，どうすっか。よくにらめっこすると，2つのパターンで繰り上がりの有無が判定できることがわかることになっている。

まず，AとBがどちらも1のとき(AND)である。で，図8-bが書ける。

もうひとつは，AかBが1 (XOR) で，なおかつ下からの繰り上がりがあるときである（AとBのXORとAND)。で，図8-c。

両者を合わせると，図9-aとなって，めでたしめでたし。

あとは，こいつをたくさんつなげれば何桁でも足し算ができるわけだ(図9-b)。

一番下の桁の下からの繰り上がりは0に固定ね。なお，アセンブラを触る人ならとっくにお見通しだろうが，この繰り上がりとは，いうまでもなくキャリフラグのことである。

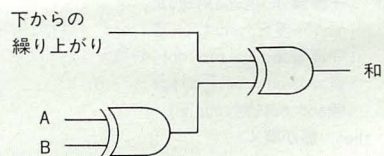
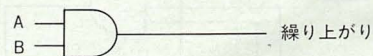
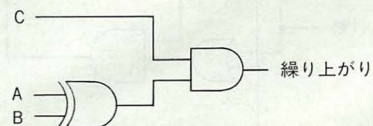
図7 足し算の真理値表

入力			出力	
下からの繰り上がり	An	Bn	和	上への繰り上がり
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

いっせいの、せ！

さて，せっかくだからこの加算器を使うことを考えよう。入力はどうしようか。とりあえず，トグルスイッチでもマイクロスイッチでもなんでもいいから，手で1桁ずつ入れる。パチパチパチ（これじゃそろばんみたいだな)。

すると，この加算器はハードで直接組まれているから，スイッチをパチパチやる端

図8 和を求める各パーツ

から出力に——たとえば発光ダイオードだとしよう——1の立った桁にはピカピカと明かりが灯っていく。当たり前だ。では，入力が終わるのを待って，いっせいにパッと答えが出るようにしたいもの。

そこで，“いっせいの，せ！　スイッチ”をつけてみる。入力をパチパチし終わったら，このスイッチを入れる。すると，答えのダイオードがピカリとつく。

答えは簡単。図10のように，全入力にANDをかませてやり，そのANDの一方をひとつにして，スイッチにつなげてやればいいのだ。ANDだから，スイッチを入れると，入力が1のところだけ加算器に1が送り込まれる。

そうである。このスイッチは加算器への入力のタイミングをとっているのだ。こういった仕組みを同期式といい，同期をとる入力をクロックパルスという。もっと回路が高速で（人間がパチパチするんじゃなくて，パソコンなんかのように速くスイッチが切り換わる），正確に動作してほしいとき，このクロックパルスには周期的にパルスを発生する，たとえば水晶発信器が使われる。パソコンの話で頻出するクロック何MHzというのは，この同期のパルスの速さなのだ。4MHzなら1秒間に4,000,000回動作を繰り返す。

電子が回路やLSIの中を駆け巡る速度というのは，いくら速いとはいえ，光速ほどではないので，長い回路だとタイミングがどこかしらで狂って，動作が不安定になる。クロックはそれを防ぐために，いっせいの，せ！　といつもやって，変なことになんないように先導しているのである。感謝感謝。だから，クロックが速いほうが回路は速いのだが，速くしすぎてほかがついてこられなくなるとハチャメチャになるので，ただクロックを上げろといってもそうは簡単にいかないのである。

図9-a　和を求める回路

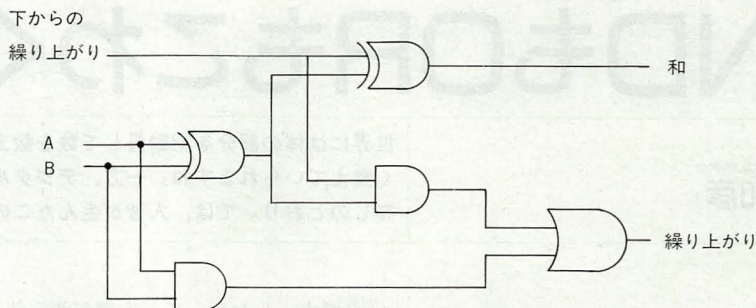



図9-b　n桁の足し算

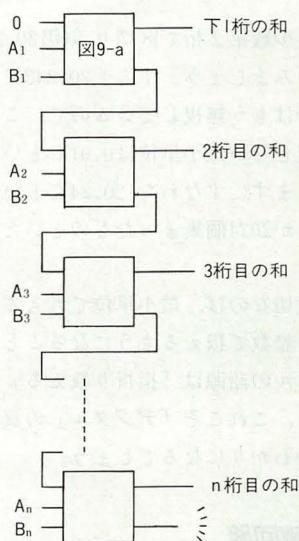
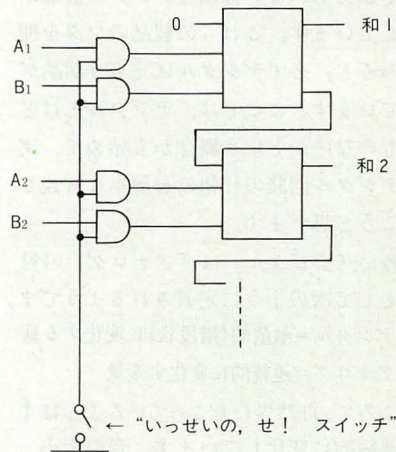



図10　“いっせいの，せ！　スイッチ”

と，いうわけであった

今までの話は，机上のものであって，実際には，前述したように電気はアナログでわがままなものなので，実際に回路を組もうと思ったら，いろいろなコツやら，芸術的な基板の使い方など大変だが，基本設計はソフトを組むのとよう似たもんだ，というのがわかっていただけただろうか。

また，デジタル回路とはいっても，ANDやらOR以外にも，“ちょっと待った！”入力を持つスリーステートやらオープンコレクタやら，物覚えのいいラッチやら，初めからクロック用の入力を持つフリップフロップなどいろいろあるが，それはまた，もう少しまじめな入門記事でということで，次からのページをお読みいただきたい。

もし，初めてICを使った工作をしようと思い立った方があれば，一度，電源（論理回路のICはわがままにも，動作するには電気をくれ，と要求するのだ）の＋と－を逆につないで，ビビッと電気を流してみるといい。すると，本当にICは煙を出して悶絶するのである。弱いやつなのだ。これは面白いから，どうせ，ICなんて安いし，一見の価値がある。死して屍拾うやつなし，命短したすきに長し，てね。

では，このへんで……え？　斎場パンクローとジョセフソン・素子(モトコ)はなんだったのかって？　それはあなた，なんだったのでしょう。また次の機会への顔見世，と，いうことで，いいじゃないですか。

〈参考文献〉

松本光功，「論理回路」，昭晃堂
ウイリアム・ギブソン，「カウント・ゼロ」，ハヤカワSF文庫
TTL IC規格表，CQ出版社

●デジタル回路入門編

デジタル回路の基礎知識

ANDもORもこわくない

Misawa Kazuhiko
三沢 和彦

世界には体の部分を総動員して数を数える人々がいます。目鼻や指など，使った部位をよく覚えていられますね。一方，デジタル回路は0と1しか区別しませんが，その可能性はご存じのとおり。では，人智が生んだこの巧妙なデバイスについての基礎を復習しましょう。

デジタルとは

私たちの身の回りでは，時計からパソコンまで数え切れない種類のデジタル製品が普及しています。これらの製品のフタを開けてみると，必ずデジタルICという部品が並んでいます。ここでは，デジタルとはどんなものなのかという概念から始めて，実際のデジタル回路の仕組みを簡単に解説していこうと思います。

一般に,「デジタル」は「アナログ」の対義語として次のように定義されるようです。

デジタル＝離散的(階段状)に変化する量

アナログ＝連続的に変化する量

ところで,自然界に起こっていることはすべて連続的に変化しています。音の大小，光の明暗，温度の高低など，みなアナログ量です。これらの自然の量を人間が測定し記録するときには，たとえば気温なら20℃というように数字で表しますね。この場合，20℃といっても19.5℃以上20.5℃未満のことですから，もっと厳密に測ると20.2℃かもしれません。さらに正確には20.24℃，あるいは20.243℃かもしれません。気温が連続的に変化する量である限り，どんなに細かく測っても完全に正確な値は求められないのです。したがって，人間が測定値を有限の時間内で処理するためには,どこか(小数第何桁か）でその値を区切らなければなりません。

この例を少数第2桁で区切り,気温20.24℃と表してみましょう。すると20.242℃か20.243℃かはもう無視しているので，この場合，無視しない最小単位は0.01℃ということになります。すなわち,20.24℃というのは0.01℃が2024個集まったものといえるわけです。

ここで大切なのは，最小単位で考える限り，すべて整数で扱えるようになることです。デジタルの語源は「指折り数える」ことですから，これこそ「デジタル」の真義であるとおわかりになるでしょう。

2値回路

デジタルの世界では数値は整数として考え直すことができ，すべての整数は2進法を使えば0と1のみで表せます。つまり，0と1さえ区別できればOKだということです。電子回路では，これをスイッチのONとOFFという2つの状態で区別するのが簡単です。

余談ですが，現代のエレクトロニクスの発展は，いかに優れたスイッチを作るかという点に絞られています。真空管にしてもトランジスタにしても，原理的にはONとOFFを区別するスイッチです。これらをより高速，小型，安価にする努力がLSI技術です。今後は電子回路に限らず,CDに見られるような光によってONとOFFを区別する技術が発展していくでしょう。

このように，スイッチのONとOFFを数字の0と1に対応させて組み合わせたものを2値回路といいます。2値回路では，数値処理が可能なだけでなく「論理演算」も可能です。論理演算とは，ある事柄についての判断を，それが当てはまるかどうか，つまりYESかNOかのふたつに分け，さまざまな条件判断を行うものです。

たとえば，パソコンユーザーについて「ハードに詳しく自分で工作もできる」という事柄Aと「プログラミングが得意で高級言語も操れる」という事柄Bを取り上げてみます。すると表1のように4通りの組み合わせが考えられます。

ストロングタイプ：AでありかつBである
$(A\cap B)$

メカトロマニア：AでありかつBでない
$(A\cap\bar{B})$

プログラマ：AでなくかつBである
$(\bar{A}\cap B)$

ビギナー：AでなくかつBでない$(\bar{A}\cap\bar{B})$

ここで記号$\bar{A}$は「Aでない」意味でNOT Aと読み，∩は上記の「かつ」にあたり$A\cap B$はA AND Bと読みます。またあとで出てき

表1　4通りのパソコンユーザー

		A　ハードが得意	
		YES	NO
B ソフトが得意	YES	誰の挑戦でも受ける **ストロングタイプ**	BASIC，C，PASCAL，果てはSLANGまで **プログラマ**
	NO	パソコンよりラジコンのほうがいいのでは？ **メカトロニクス・マニア**	これからがんばります **ビギナー**

表2　条件判断

		A　ハードが得意	
		YES(A＝1)	NO(A＝0)
B ソフトが得意	YES(B＝1)	ストロング　$A\times B=1$ メカトロ　$A\times\bar{B}=0$ プログラマ　$\bar{A}\times B=0$ ビギナー　$\bar{A}\times\bar{B}=0$	$A\times B=0$ $A\times\bar{B}=0$ $\bar{A}\times B=1$ $\bar{A}\times\bar{B}=0$
	NO(B＝0)	$A\times B=0$ $A\times\bar{B}=1$ $\bar{A}\times B=0$ $\bar{A}\times\bar{B}=0$	$A\times B=0$ $A\times\bar{B}=0$ $\bar{A}\times B=0$ $\bar{A}\times\bar{B}=1$

表3　排他的論理和

Yは演算式$(A\times\bar{B})+(\bar{A}\times B)$の答え

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

ますが，A∪Bは「AまたはB」という意味でA ORBとも読みます。

ここで，事柄AとBは，YES＝1, NO＝0のどちらかを値として取れる変数だと考えます。また，ANDは掛け算，ORは足し算に対応させてみましょう。するとストロングタイプの場合，A＝1，B＝1を代入したときだけ，A∩B(A×B)＝1になります。メカトロマニアを表すA∩$\bar{B}$(A×$\bar{B}$)は，A＝1(YES)，B＝0(NO)のときだけ1になります(B＝1のとき$\bar{B}$＝0，B＝0のとき$\bar{B}$＝1と計算してください)。メカトロマニアにA＝1，B＝1を代入するとA×$\bar{B}$＝0です。そこで，この掛け算の答(論理積)が1のときは条件成立，0のときは不成立と約束すれば，1と0のふたつの値を使って条件判断ができることになります（表2)。

次に，ハードまたはソフトの少なくともどちらかができる人（ストロングかメカトロかプログラマか）の条件を式に表してみましょう。これはA∪Bとなり，演算式はA＋Bです。このとき，A＝1，B＝1のストロングタイプではA＋B＝2になってしまいますが，値は0か1のどちらかなので，このときもA＋B＝1と約束します。これを論理和といって普通の足し算と区別しています。

さらに，ハードかソフトのどちらか一方のみ得意という中級者を表してみます。この場合，メカトロマニア(A×$\bar{B}$)またはプログラマ($\bar{A}$×B)なので演算式は，

(A×$\bar{B}$)＋($\bar{A}$×B)

となります。再び表2を見てください。A×$\bar{B}$あるいは$\bar{A}$×Bが1の値を取るのは，それぞれA＝1でB＝0のときと，A＝0でB＝1のときだけです。このように，A＝1，B＝1の場合も，つまりAとBの値が両方とも1か0をとるとき条件不成立となるものを，排他的論理和といいます(表3)。

以上のように，事柄に対するYES，NOを0，1で表し，条件判断を演算で行う方法の体系を論理代数(この場合2値論理代数)といいます。詳しくは参考文献などを見ていただくとして，ここでは，0と1を扱うデジタル回路が2進数の数値処理だけでなくYES/NOの条件判断にも利用できることを覚えておいてください。

デジタルICと基本ゲート回路

では，数値処理と論理演算を実際の電子回路がどのように行っているかの解説に移りましょう。実は2進数の計算も論理演算の組み合わせでできるので，主題は論理回路の仕組みということになります。

現在，論理回路はデジタルICを組み合わせて作られています。デジタルICは，一見したところ足が何本か出ている小さな黒い箱ですが，中身はトランジスタやダイオー

図1　NOT回路LS04

a)

b)

図3　OR回路LS32

a)

b)

Y＝A＋B

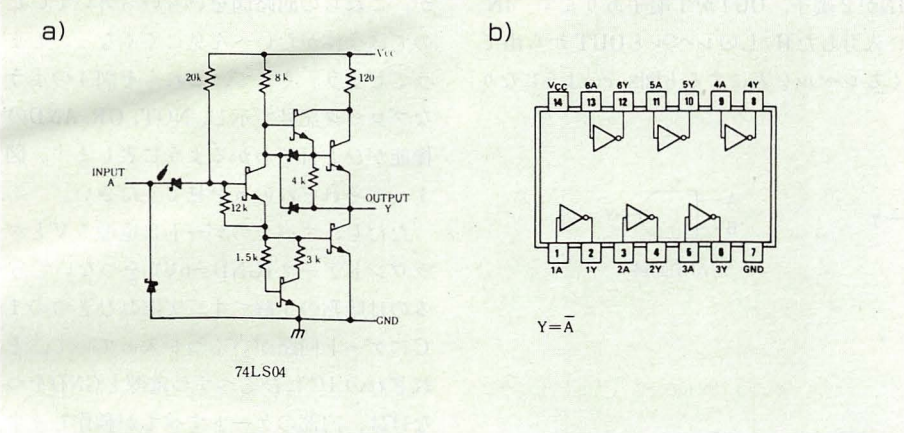
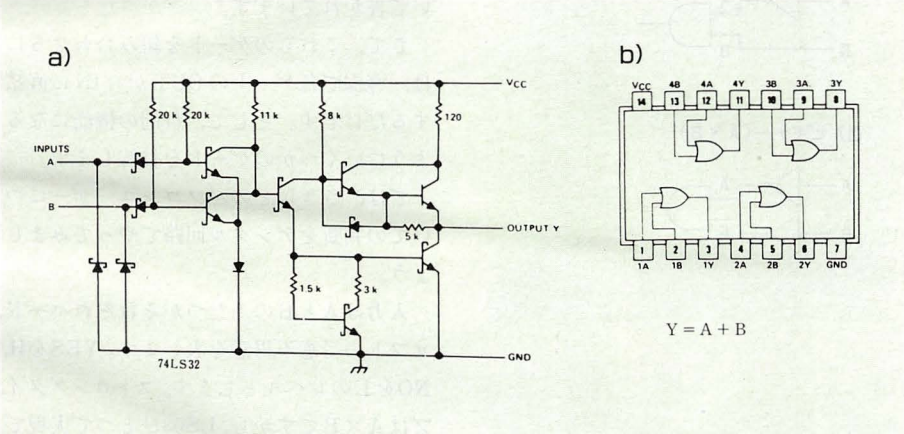
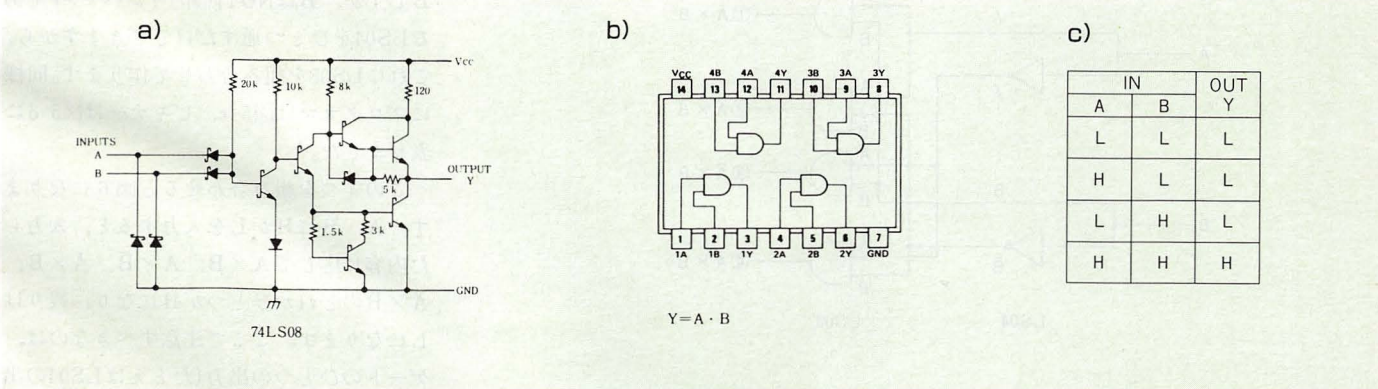
図2　AND回路LS08

a)

b)

Y＝A・B

c)

IN		OUT
A	B	Y
L	L	L
H	L	L
L	H	L
H	H	H

ド，抵抗，コンデンサなどの部品がびっしり並んだ電子回路です(図１)。デジタルＩＣはその電気的特性によってTTLとC-MOSというふたつのグループに大別できます。それぞれ特長がありますが，ここではマイコン回路によく使われるTTLファミリについて説明しましょう。

TTL ICは５Ｖの電源をつないで動かします。図１-aの回路図はLS04という型番のNOT回路(インバータという)です。この回路にはIN(入力)，OUT(出力)の端子がひとつずつついていて，それぞれがICの足となって出ているわけです。

さきほどの２値回路でも触れましたが，NOTは１を０に，０を１に反転させる演算です。NOT回路では，INに５Ｖをつなぐと OUTに０V, INに０VをつなぐとOUTに5Vが出るようになっています。すなわち，5Vを１，０Ｖを０に対応させているのです。

このように，デジタルICは端子の電圧にHigh(5V)とLow(0V) のふたつのレベルを持つ論理演算素子です。ただ，実際には入力電圧が2.0Ｖから5.0ＶならＨ(High)，0.0Ｖから0.8ＶならＬ(Low)と判断するようになっているので，電圧値に多少の幅があっても処理できます。

図2-aはTTLのLS08という型番の回路で，INが２端子，OUTが１端子あります。INに入力したH/LのレベルとOUTから出てくるレベルを表にすると図2-cのようになりますが，Ｈを1，Ｌを0と書き直すと，INへの入力が両方とも１のときのみOUTも１になることがわかるでしょう。これをAND回路といいます。また，図3-aはLS32というOR回路で，やはりINを２端子，OUTを１端子持ち，INがどちらか一方でも1になればOUTも１になります。

これらの回路をゲートと呼びます。ゲートとは，ご存じのように門という意味で，この門を通ると論理演算が行われるわけです。

どんなデジタル回路でもNOT, OR, ANDの基本ゲートの組み合わせで実現できますが，これらの回路図をいちいち書いていたのでは全体がたいへん見にくくなってしまうでしょう。そこで，これらを図４のようなブロック記号で示し，NOT，OR, ANDの機能がひと目でわかるように表します。図１～３それぞれのｂを見てください。

ただし，すべてのゲートに電源５Ｖとグラウンドアース(GND＝0V) をつないであるのは暗黙の了解です。実際はひとつのＩＣにゲート回路がいくつか入っていて，それぞれのICにひとつずつ電源とGNDをつなげば，内部のゲートすべてが動作します。一般の回路図は，このブロック記号をつないで書かれています。

さて，これらのゲートを組み合わせるには，導線で各ゲートのOUTからINに直結するだけです。そして，目的の機能になるようにいくつかのゲートを結線します。

では，さきほどのパソコンユーザーについての判断をデジタル回路でやってみましょう。

入力はＡとＢのふたつがそれぞれハード，ソフトの得意不得意を表します。YESをH, NOをＬのレベルとします。ストロングタイプはＡ×Ｂですから，LS08ひとつで実現できます(図５a)。次にメカトロマニアは$A\times\bar{B}$ですが，$\bar{B}$はNOT回路(インバータ) であるLS04をひとつ通すだけでできますから，これにLS08を組み合わせて作ります。同様にプログラマは図5-c，ビギナーは図5-dに表しました。

この４つを組み合わせると図６になります。Ａ，ＢにＨかＬを入力すると，入力した内容に応じて$A\times B$，$A\times\bar{B}$，$\bar{A}\times B$，$\bar{A}\times\bar{B}$のどれかひとつがＨになり，残りはＬになります。ここで注意すべきなのは，ゲートのひとつの出力(たとえばLS04の出

図４　ブロック記号(MIL記号)

図５　論理回路で表すと

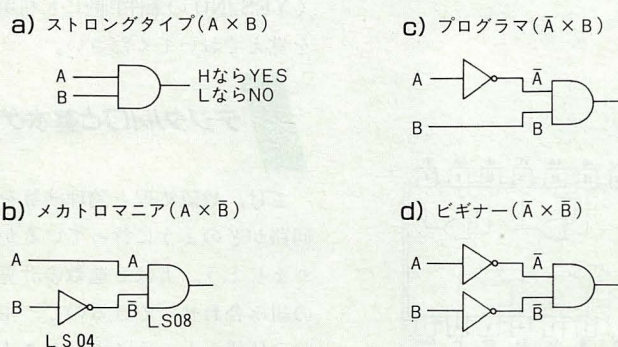
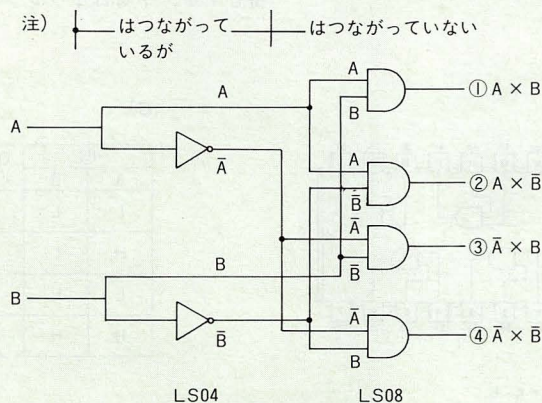



図６　パソコンユーザー判定器

力Ā）から２つ以上のゲートの入力(LS08の③，④の入力）につなぐことができるという点です。こうすると，２つ以上のゲートを同時に動かせます。ただし，TTL-LSファミリではひとつの出力につなげるゲートの数は10個までです。

では，例題としてハード，ソフトのどちらか一方のみ得意という(A×B̄)＋(Ā×B)の中級者について判断する回路を組んでみてください。解答は図７に示しておきます。

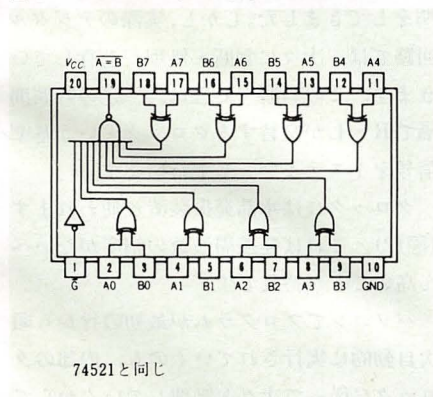
ゲートを組み合わせた回路

NOT，OR，ANDを組み合わせるとどんな回路でも作れますが，実際にこれらの基本ゲートをその都度組み合わせていくとコストも労力もたいへんなものになります。そこでICの数を少なくするために，TTLファミリには，よく使われる機能を組み合わせてひとつのICに入れたものが多種あります。ですから，ICを使うときはどんな機能のものがあるか知っておくことが大切です。

市販されているTTL IC規格表を手に入れると，現在どのようなICが出回っているか調べることができます。たとえば，LS139というICが，さきほど作ったパソコンユーザー判定器と同じ機能を持つことがわかります。LS04とLS08を結線した回路がLS139をひとつ持ってきただけで実現されてしまうのです(図８)。ただし，LS139では，LをYES，HをNOとしなければならないので注意が必要です。

ENABLE(イネーブル)Gというのは全体のスイッチで，ここをHにするとOFF，LにするとONになります。また，例題として挙げた中級者判定器はLS86ひとつで実現できます。

図９　コンパレータLS688

74521と同じ

さて，規格表を見るときの参考のために，よく使われる機能を簡単に説明することにしましょう。

デコーダ

パソコンユーザー判定器のように，いくつかの入力の組み合わせから特定のものを判定する回路をいいます。代表的なのは，３桁の２進数を8通りに判定するLS138と，前述したLS139(これも２桁の２進数を４通りに判定するものと考えてよい)です。n桁の２進数はn bit(binary digit) といいます。また，10進数の０から９に相当する２進数0000Bから1001Bまでを９通りに出力するBCDデコーダLS42というものもあります。

エンコーダ

デコーダの逆の機能で，たとえば８通りの入力を３桁の２進数に変えるLS148，10通りの入力を0000Bから1001Bの２進数に変えるLS147などがあります。

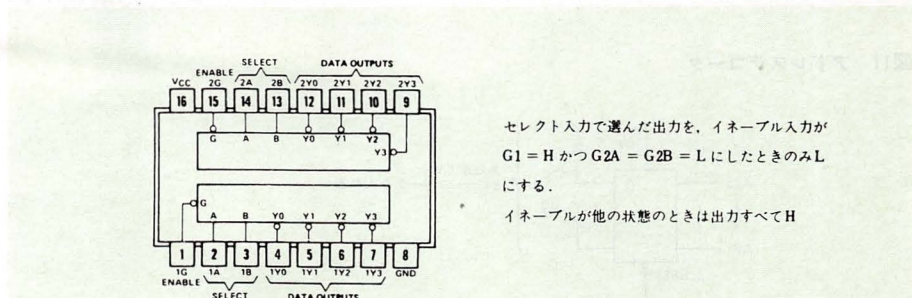
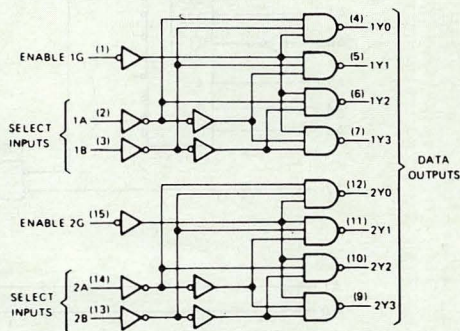
コンパレータ

これはあらかじめ設定した２進数のデータと入力されたデータが等しいか否かを判定するものです(図９)。たとえばLS86は，ふたつの入力が等しいとL，等しくないとHを出力する１桁のコンパレータで，Ex. OR(エクスクルーシブ・オア)回路ともいいます。これを8桁組み合わせたものがLS688ですが，これは，すべての桁が等しくないときだけLが出力されるので，データの一致を検出したいときは，あらかじめ与えるデータはすべてNOTで反転させておかなければなりません。

図７　中級者を判定する回路

図８　デコーダLS139

セレクト入力で選んだ出力を，イネーブル入力がG1＝HかつG2A＝G2B＝Lにしたときのみ Lにする．

イネーブルが他の状態のときは出力すべてH

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

データセレクタ

これは，いくつかのデータからひとつを切り換えて取り出すスイッチです（図10）。LS151は，D0からD7に入力した８通りのデータから３桁の２進数で指定したデータを選択して出力するものです。デコーダが０〜７のどれかが選択されるとそこだけ必ずLが出るのに対し，こちらは入力したデータがHならH，LならLが出ます。

加算回路

２進数の足し算を繰り上がり（キャリ）ありで行います。LS83は４ビットの加算器で，出力は４ビット＋キャリの５端子あります。

以上が代表的な組み合わせデータ回路です。では，まとめとして実際のパソコンで使われている例を挙げてみましょう。インタフェイスボードなどのI/Oアドレスを判定する回路（アドレスデコーダという）です。

I/Oアドレスというのは，16ビットの数でこのアドレスを指定して，特定のインタフェイスにアクセスするものでしたね。そこで，特定のアドレスをあらかじめデータとして持っていて，入力したアドレスと一致したときだけ出力があるような回路が必要なわけです。図11がその回路です。この回路はそのまま使えますので試してみてください。私は自作のX1用MIDIインタフェイスボードに同じ回路を使っています。本誌掲載のMIDIボードのアドレスデコーダ部もほとんど同じです。

この回路の動作を説明しましょう。まずLS688では，あらかじめデータとしてすべてHを与えているので，A15からA8の８ビット（A15はアドレスの15桁目の意味）がすべてLのときだけ出力にLが出ます。そしてLS138では，A15からA3がすべてLのときだけ出力がLになる端子から出力を取っています。LS32ではさらにA2もLでないと出力がLになりません。

ところで，LS32はORゲートなのに前述したものとは違う記号が使われています。ORゲートの性質，つまり「少なくともどちらか一方がHならH」ということと「両方ともLのときだけL」ということは同じなのですが，特に前者の意味で使うときはと書き，後者（この場合）の意味ではと書くのです。これは正論理，負論理の違いといいます。

要するに，A15からA2のすべてがLのとき初めて，LS139がONになります。LS139がONになれば，A1とA0の2ビットのアドレスが４通りに出力されるので，これで0000Hから0003Hを判定することができるわけです。

図10　データセレクタLS152

INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	S	Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

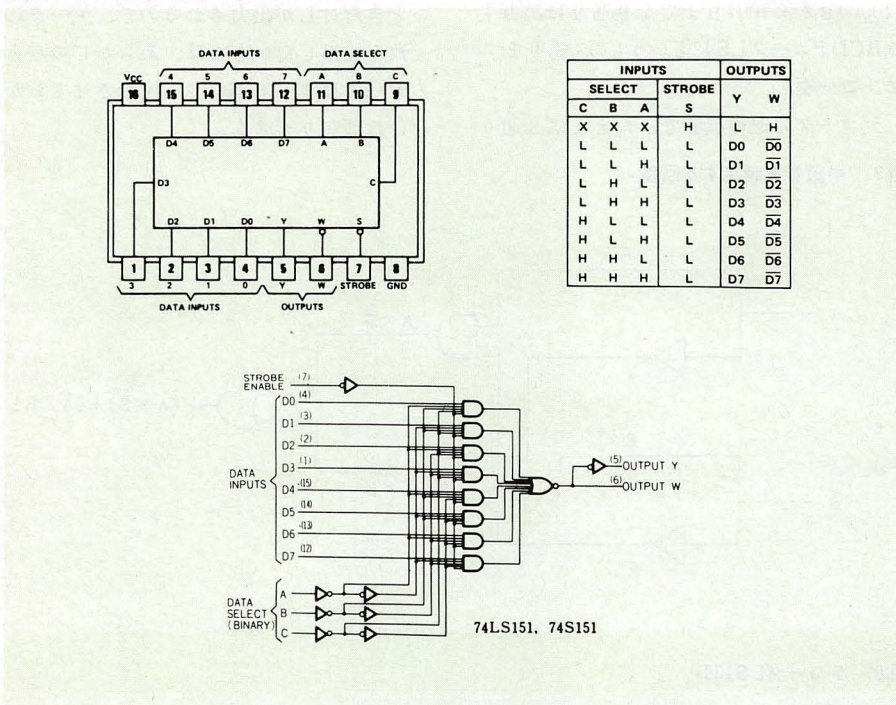
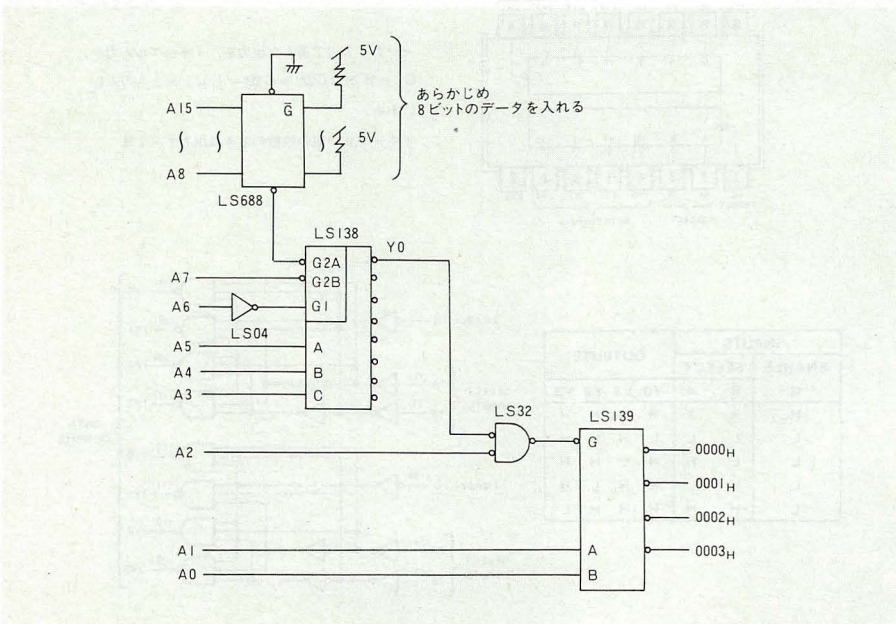



図11　アドレスデコーダ

クロックとラッチ

これまで，扱う信号の入出力が定常である（安定している）場合の回路について説明をしてきました。しかし，実際のデジタル回路では，次々に判断・処理が進行していきます。この判断・処理は，一定の時間間隔でHとLが交替するクロックという基準信号を１ステップとして行います。

クロックには水晶発振装置が使われます（図12）。これは発振周波数の精度がたいへん高いことが特長です。

パソコンでプログラムが最初の行から順次自動的に実行されていくのも，内部のクロックに従って次々と処理していくからで

す。ちなみにX1は4MHz，X68000は10MHzのクロックを持っています。

次に，クロックを持った回路が外部からの信号を読み取ることを考えてみましょう。図13を見てください。この図によると，クロックの立ち上がりで読み取りを行いますが，もし立ち上がりと立ち上がりの間に外部信号が来てしまったら，それは検出できないことになります。そこで，たとえば周期の間にHの信号が来たら，そのあともずっとHを保持しておく回路が必要になります。この保持回路をラッチといいます。これもゲートの組み合わせで実現できます。

図14を見てください。これはLS00というゲート(NAND＝NOT AND)を2個分組み合わせたものですが，片方の出力がもう一方の入力に入り，その出力がもとの入力に入ってフィードバックされています。ここで，$\overline{R}$をHにしたまま$\overline{S}$をLにしてみると，QはHになるので，$\overline{Q}$はLになります。ここで，$\overline{S}$をHに戻しても，$\overline{Q}$がLなのでQはHのまま，$\overline{R}$もHのままなので$\overline{Q}$はLというように矛盾が生じません。つまりQがH，$\overline{Q}$がLの状態を保持しているわけです。

次に$\overline{S}$をHにしたままで$\overline{R}$をLにすると，①と②をひっくり返したのと同じことになり，今度はQがL，$\overline{Q}$がHに反転します。このとき$\overline{R}$をHに戻してもやはり状態を保持します。こういう機能のラッチをR-S(Reset-Set)ラッチといいます。LS279が専用のR-Sラッチになっています。

次にR-Sラッチの発展型でより実用的なDラッチについて説明しましょう。Dラッチには，入力としてD(データ)とSTB(ストローブ)のふたつがあります(図15)。Dには時々刻々と変化するデータが入ってきます。STBがHのときはそのまま出てきますが，STBをLにすると，その瞬間のDの状態をそのまま保持します。このラッチは，任

図12 水晶発振回路

図13 外部信号がきたとき

図15 Dラッチ

図16 タイムチャート

図14 R-Sラッチ

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
L	L	許されない	
L	H	H	L
H	L	L	H
H	H	前の状態を保持	

NAND

A	B	出力
L	L	H
L	H	H
H	L	H
H	H	L

意のデータをある瞬間で止めて見るときに使われます。この様子をタイムチャートと呼ばれる図に表してみましょう(図16)。タイムチャートは，左から右に時間が進んでいく図で，時間と共に状態が変化していく様子がわかりやすいようになっています。

実際にはDラッチよりもDフリップフロップ(D-FF)が多く使われるようです。このラッチとフリップフロップは混同されがちですが，囲みに挙げた解説を見て違いを把握しておいてください。

さて，D-FFは状態を保持することからもわかるようにメモリそのものです。幅広く応用でき，世の中のデジタル回路でこれを使っていないものはないと言っても過言ではありません。

身近な例としてパソコンのインタフェイスをとって考えてみましょう。CPUをはじめとするパソコン内のシステムは，すべてデータバスという信号線 (8ビットのCPUならパラレル8本)を持ち，そのバスを通じてメモリなどの周辺チップとデータのやりとりを行います。インタフェイスを通じて外部とのデータの入出力を行うときも，データはバスに乗って流れます。しかし，バス上は常にさまざまなデータが入れ替わり流れているので，外部に出力するのもそのままでは一瞬だけになってしまいます。外部の機器にとってCPUがデータをバスに出したと同時に読み取るのはとてもタイミングが厳しいので，外部に出力するデータを保持しておかなければなりません。それにはD-FFが8個パラレルに入っているLS273(図17)が便利なので，これを使って回路を組んでみました(図18)。

I/Oにデータを出力するときには，CPUはアドレスを指定したあとでバスにデータを送り出し，$\overline{\mathrm{IORQ}}$と$\overline{\mathrm{WR}}$を同時にLにします。タイミングチャートを見てください(図19)。LS273はクロック(CK) 入力がLからHに立ち上がるとき，Dに入力されているデータをラッチします。タイミングチャートを見てもわかりますが，この回路は，デコーダLS139を使ってアドレスデコーダに指定したアドレスが入り，かつ$\overline{\mathrm{IORQ}}$と$\overline{\mathrm{WR}}$が同時にLになるとき(I/O OUTのとき) にCKにLがくるようになっています。したがって，I/O OUT命令が実行されたときのデータがラッチされることになります。 I/O OUTが実行されない限り，ラッチはデータを変えません。このインタフェイスがあって

図18 インタフェイス

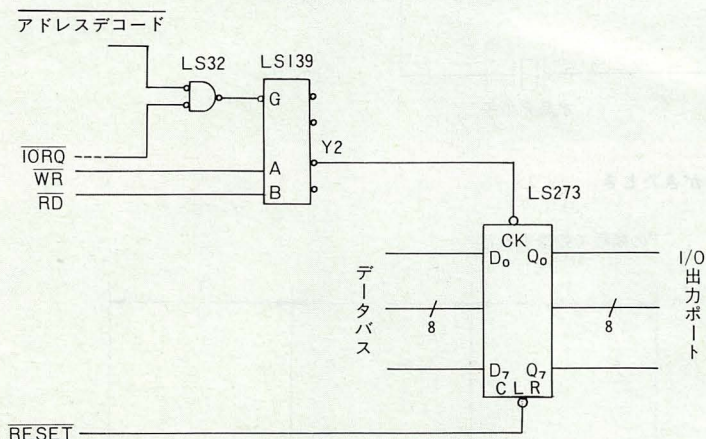
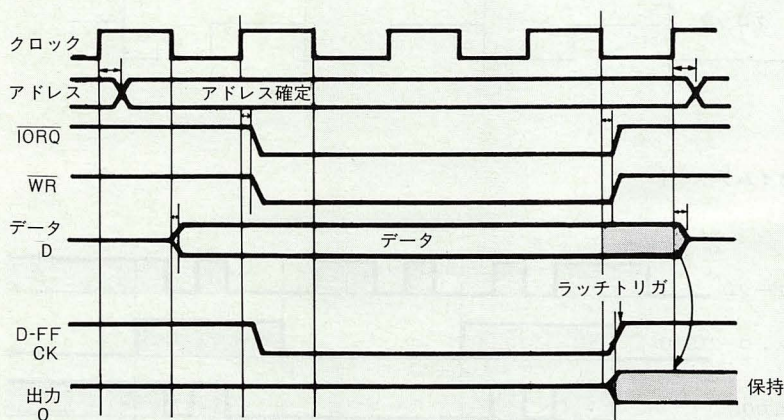
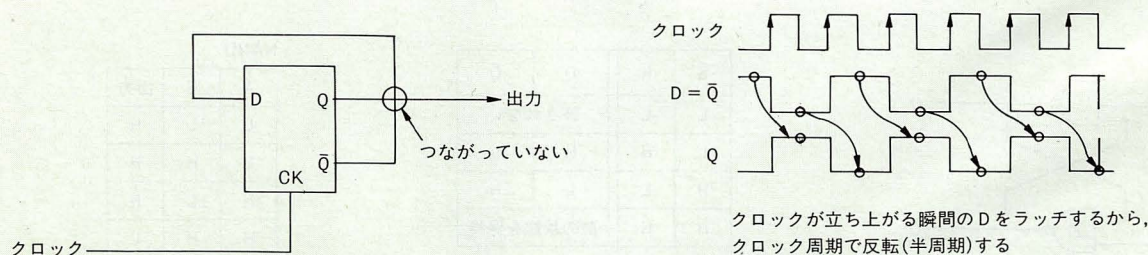



図19 タイミングチャート

図20 D-FFの動作

クロックが立ち上がる瞬間のDをラッチするから，
クロック周期で反転(半周期)する

初めて，外部機器はCPUのタイミングを気にせず着実にデータを読み出せるのです。ところでアドレスデコード信号には図18のデコーダをそのまま使えるので，基本ICだけでパラレルインタフェイスが組めます。

カウンタとシフトレジスタ

カウンタもシフトレジスタもD-FFをつなぎあわせたものにすぎません。ここではD-FFの動作の復習をしながら,この２つの機能について考えていきます。

D-FFは，CK入力がLのときはDに入力されたデータがそのままQに，また反転されて$\bar{Q}$に出てきます。その間はデータが変化すると出力Q，$\bar{Q}$も変化します。そしてCKをLからHに上げたとき(エッジトリガ)に，そのとき入力されていたデータを保持し，それ以降は再びCKをLからHに上げるときまで，Q，$\bar{Q}$に保持したデータを出力し続けます。そこで，図20のように結線してCKにクロックを入れてみます。すると，クロックが２周期入って初めて出力が１周期出ることがわかります。すなわち，この回路はクロックを２個ずつカウントする機能を持っているわけです。こうした数を数えるデジタル回路をカウンタと呼びます。

デジタルの世界では，２進数で数を表現していましたね。２進数とは２つずつまとめて桁上げする仕組みなのですから，D-FFのカウンタでいえば出力が桁上げ信号にあたるわけです。したがってこのカウンタを４個直列につなげると，４ビットのカウンタになります。LS293はこれだけで４ビットカウンタとして使えます。

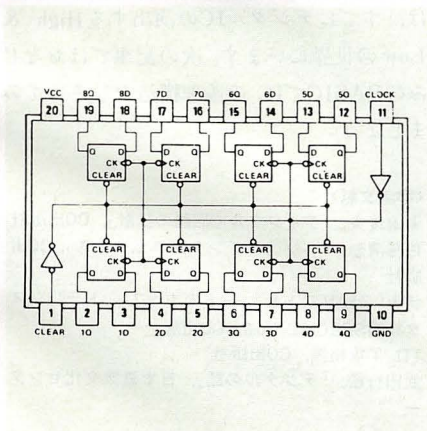
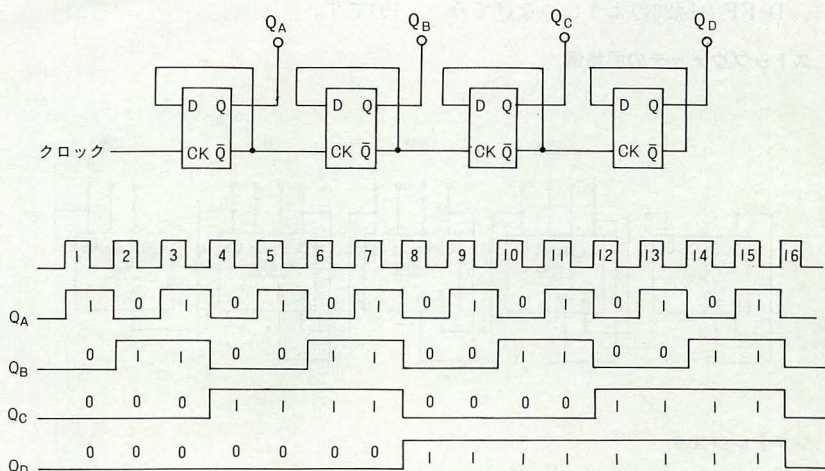
図21で，$Q_DQ_CQ_BQ_A$の順に並べてみると，クロックのカウント数を４ビット数で表しているのがわかるでしょう。

また，日常使われるのは10進数なので，10カウントで桁上げ信号を出すカウンタLS290も使われます。LS290では，２進数で0000_Bから1001_Bを出力して，次には桁上げを出すと同時に0000_Bに戻ります。

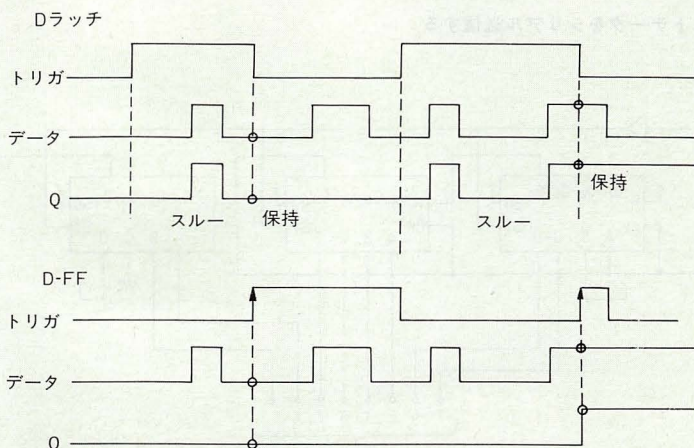
図22はストップウォッチの回路例です。左端のLS290は単純な1/10分周回路で，この回路をひとつ通すたびにクロックが１桁下がります。LS290の４つの出力を$Q_DQ_CQ_BQ_A$の順で２進数として考えると０から９を表しています。これを表示回路（ディスプレイデコーダ）につなぐとLED表示もできます。また，10秒の桁は６で桁上げなので，６進カウンタLS92を使っています。このように，時間計測というのは規則正しく出るクロックをカウントして行っているのです。

図21　４ビットカウンタ

図17　８ビットD-FF LS273

ラッチとフリップフロップの違い

このふたつはどちらも時間的に変化するデータを一時的に保持するものですが，その保持の仕方に違いがあります。

どちらにもデータ保持をコントロールするための信号としてトリガがありますが，Dラッチは，トリガがHのときはデータがそのままQに出力され(スルー状態)，トリガがLになった瞬間のデータが保持されます。したがって保持されるのはトリガがLの間だけです。それに対し，Dフリップフロップは常に保持状態にあります。そして，トリガがL→Hに立ち上がる瞬間（エッジ）のデータが保持され，それはトリガをLに戻しても変化しません。

このように，データのスルー状態があるのがラッチ，常に保持状態しかなくエッジの立ち上がりだけで保持データが切り換わるのがフリップフロップなのです。

正確な1kHzというのは，正確に1秒あたり1000個のクロックパルスを出しているわけで，これにはさきほど述べた水晶発振装置が使われます。デジタル時計をクォーツ（石英）と呼ぶのはそのためで，アナログ表示の時計でもクォーツといっているものは，時計の仕組みがデジタルなわけです。

ただ，現在は時計専用の便利なICが安く出回っているので，このようにカウンタをいくつも並べなくてもIC 1個で実現できます。

次に，D-FFを図23のようにつなげてみます。クロックが1個入るごとに各D-FFはひとつ左側のD-FFの出力を取り込みます。すなわち，データを1ビットずつ右へシフトしていくことになります。もしクロックと同期して4ビットのデータをシリアルに送ると，4個のクロックのあとで$Q_AQ_BQ_CQ_D$にそのシリアルデータが取り込まれるわけです。このように，クロック1個ごとに1ビットずつデータがシフトしていく機能のものをシフトレジスタといいます。4ビットひとまとまりになっているのがLS194です。

図22　ストップウォッチの回路例

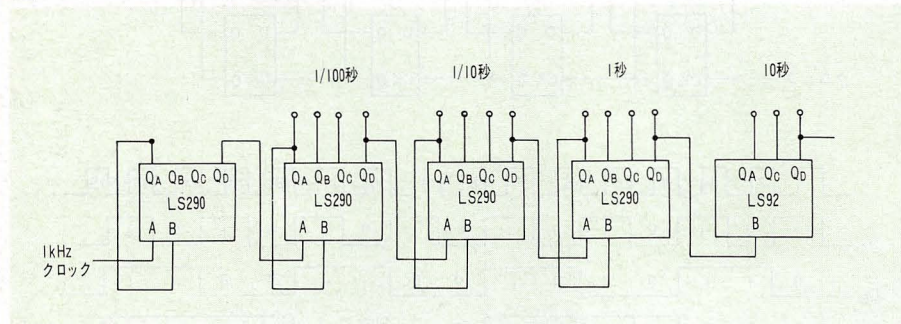



図23　シフトレジスタ

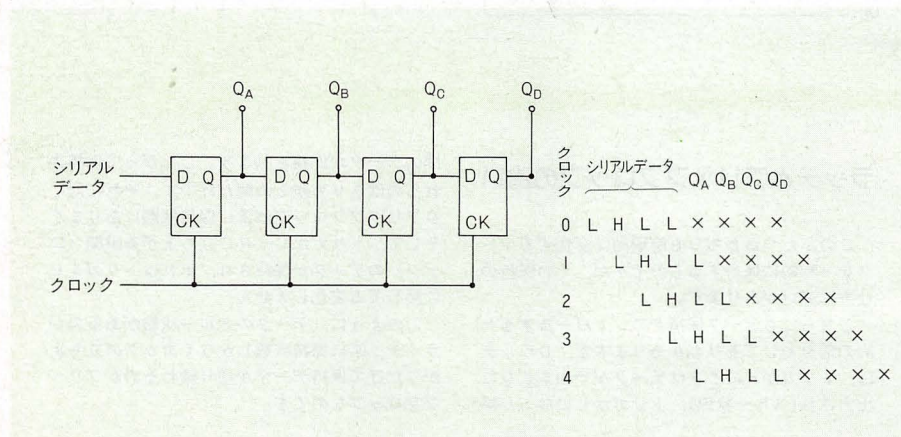
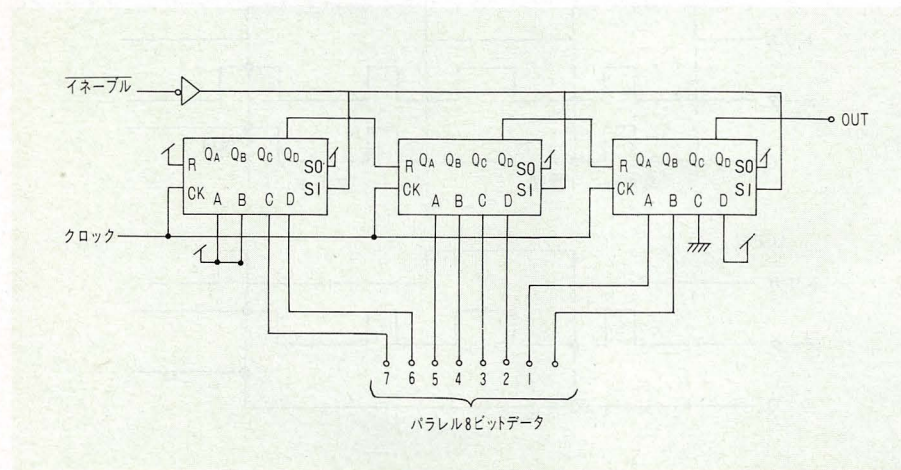



図24　8ビットデータをシリアル送信する

例として，パラレル入力した8ビットデータを，スタート1ビット（L），ストップ1ビット（H）をつけて計10ビットをシリアル送信するインタフェイスを図24に挙げて解説しましょう。LS194はR入力が右シフトの桁上げ入力になっていて，複数個つなぐことができます。また，S0をHにしたままでS1をHにするとパラレルデータのロード，Lにするとビットシフトの機能になります。通常はイネーブルをHにしておくと，OUTは常にHになっています。送信するときには，パラレルデータをセットしてからイネーブルをLにするとデータがロードされ，再びHに戻すとそのデータはシリアル送信されます。送信速度は入力クロックで決まります。

この回路はパソコンのインタフェイスとして使うときには，イネーブルにアドレスデコード信号をつなげばよいのですが，送信クロックをCPUのクロックと変えることができません。実際4MHzのシリアル通信は速すぎて信頼性が落ちるので，若干回路を変更する必要があります。

専用LSIによる未来

現代はデジタル技術の進歩が目覚ましく，家電製品のほとんどがマイコン制御になってきました。そのため，これまで述べてきたような基本ICを組み合わせた機能回路もひとつのICとして生産され，利用されています。デジタル時計用IC，電卓用IC，パソコンパラレルインタフェイスIC，シリアルインタフェイスICなど外付部品も少なく，初心者が工作するのも簡単になってきました。

さて，特集をここまで読み進んだあなたは，すでにデジタルICの演出するHigh & Lowの世界にいます。次の記事ではおなじみのBASICでICの論理構造をなぞってみましょう。

〈参考文献〉
湯山俊夫，「ディジタルIC回路の設計」，CQ出版社
佐藤清忠，「パソコンインターフェイス考」，CQ出版社
特集「Z80ソフト＆ハードのすべて」，トランジスタ技術SPECIAL No.6，CQ出版社
TTL IC規格表，CQ出版社
武田行松，「デジタルの話」，日本電気文化センター

●デジタル回路入門編

ソフトでハードをシミュレート

BASICでわかる論理回路

Shimada Atsushi
島田 淳史

電気の流れるハードウェアはとかく異質なものと敬遠されがちですが，ICとICの組み合わせを考えるのはプログラムを組むのと変わりありません。ここでは，簡単な論理回路をBASICでシミュレートしながら，ハードウェアの基本を学ぶことにしましょう。

かなりパソコン歴の長い人でも「ハードはどうも苦手だ」とか，「回路図を見ても何が書いてあるのか，さっぱりわからない」という声はよく聞かれます。しかし，ここで断言しておきましょう。「デジタル回路を理解することは，BASICで書かれたプログラムを解析することとまったく変わらない」のです。それなのに，ハードが苦手という人が続々出てくるのは，「なぜハードが難しく見えるのか」という問題を正面から取り上げた入門記事がほとんどなかったからなのです。だから，これからハードウェアの世界に入っていこうとする皆さんは，ハードは難しいという先入観をこの記事を読むことで取り払ってから，次のステップに進むのが最も近道となるでしょう。

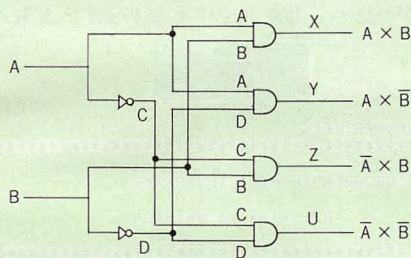
BASICの場合

それでは，皆さんの身近なBASICの話から始めましょう。BASICは英語をベースにした命令語から成り立っています。ごく簡単な例は次のようなものです。

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT
```

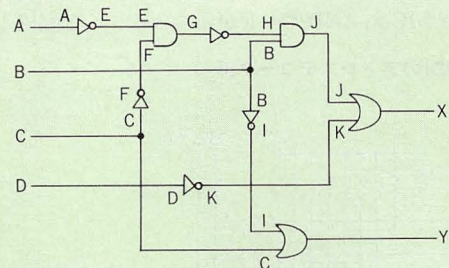
しかしパソコンが英語を理解するわけではなく，各命令語は，その処理を行う機械語のサブルーチンを持っています。テキストのFORという命令のところにくると，その処理に必要なサブルーチンを呼び出して実行するわけです。とはいえ，私たちユーザーは実際にマシン語ルーチンがどうなっているか知らないことのほうが多いでしょう。PRINT Iにしても，「Iという変数に数値を格納してからPRINTという命令を実行する

図1　パソコンユーザー判定器

図2　ちょっと複雑な回路例

リスト1

```
10 ' SAVE "1:LOGIC1.BAS
20 '*************************************************
30 '
40 '   COMPUTER USER CHECKER / DECODER
50 '
60 '               63.11.23   A.SHIMADA
70 '
80 '*************************************************
90 PRINT" A  B : X  Y  Z  U"
100 '
110 FOR A=-1 TO 0
120 FOR B=-1 TO 0
130 '
140 C=NOT A
150 D=NOT B
160 X=A AND B
170 Y=A AND D
180 Z=C AND B
190 U=C AND D
200 '
210 PRINTA;B;":";X;Y;Z;U
220 NEXT B
230 NEXT A
```

実行結果

```
RUN
 A  B : X  Y  Z  U
-1 -1 :-1  0  0  0
-1  0 : 0 -1  0  0
 0 -1 : 0  0 -1  0
 0  0 : 0  0  0 -1
Ok
```

リスト2

```
10 ' SAVE "1:LOGIC2.BAS
20 '*************************************************
30 '
40 '   COMPLICATED GATE CIRCUIT
50 '
60 '               63.11.23   A.SHIMADA
70 '
80 '*************************************************
90 PRINT" A  B  C  D : X  Y"
100 '
110 FOR A=-1 TO 0
120 FOR B=-1 TO 0
130 FOR C=-1 TO 0
140 FOR D=-1 TO 0
150 '
160 GOSUB "LOGIC"
170 '
180 PRINTA;B;C;D;":";X;Y
190 NEXT D
200 NEXT C
210 NEXT B
220 NEXT A
230 END
240 '
250 LABEL "LOGIC"
260 E=NOT A
270 F=NOT C
280 G=E AND F
290 H=NOT G
300 I=NOT B
310 J=H AND B
320 K=NOT D
330 X=J OR K
340 Y=I OR C
350 RETURN
```

実行結果

```
RUN
 A  B  C  D : X  Y
-1 -1 -1 -1 :-1 -1
-1 -1 -1  0 :-1 -1
-1 -1  0 -1 :-1  0
-1 -1  0  0 :-1  0
-1  0 -1 -1 : 0 -1
-1  0 -1  0 :-1 -1
-1  0  0 -1 : 0 -1
-1  0  0  0 :-1 -1
 0 -1 -1 -1 :-1 -1
 0 -1 -1  0 :-1 -1
 0 -1  0 -1 : 0  0
 0 -1  0  0 :-1  0
 0  0 -1 -1 : 0 -1
 0  0 -1  0 :-1 -1
 0  0  0 -1 : 0 -1
 0  0  0  0 :-1 -1
Ok
```

論理式 1

```
DEF FNYn=
NOT(((G1 AND (NOT G2A)) AND (NOT G2B)) AND (4×C+2×B+A=n))
```

（※原文では最後の G2B に上線あり：$\overline{\text{G2B}}$）

論理式 2（BASIC）

```
LAVEL "Yn"
    IF G1=-1 AND G2A=0 AND G2B=0
        THEN IF 4×C+2×B+A=n THEN Y=0 ELSE Y=-1
        ELSE Y=-1
RETURN
```

と，画面上にＩの内容が表示される」という機能を外から知っているだけです。

これは，デジタルICの世界でも事情はまったく同じなのです。たいていのユーザーはチップの中で実際にどのように電流が流れているかなど知りません。ブラックボックスの外からどのような入力を与えれば，箱の外にどう出力されてくるかという機能を知ってさえすれば十分なのです。

ICは関数だ

たとえば，荻窪氏の記事にも登場しているLS08というICは，ANDゲートが４つ入っています。ANDゲートとは，A，Bの２つの入力に+5V(H)か0V(L)かをつなぐと，出力にもHかLが真理値表に従って現れる機能を持ったものです。ここで，N→－１,L→０と対応させれば，BASICの命令でそのまま，

　Y＝A AND B

と書けます。

BASICでも，ANDという演算子は機械語ルーチンでどうなっているか知らなくてもよいわけで，同様に，

　OR　→　Y＝A OR B
　NOT　→　Y＝NOT A

とBASICで表せます。

では，三沢氏の解説で出てきたパソコンユーザー判定器を考えてみましょう（図１）。ひとつのゲートは，それぞれのひとつの演算子として表せます。そして，つないである配線１本にひとつの状態（HかL）が対応するので，配線ごとに論理変数（A，B，C，D，X，Y，Z，U）を対応させます。入力A, BをINPUTにして，出力X～UをPRINTさせるBASICプログラムをリスト１に示しました。

次に，もっと複雑な例を図２に示します。この場合も，１本の結線につきひとつの変数を割り当て，順番に計算していくと真理値表が得られます。プログラムはリスト２に示しました。サブルーチン“LOGIC”内が論理計算です。

以上のように，ICをブラックボックスとして考え，入力引数を与えて出力値を得る関数(命令)だと割り切ってしまえば，BASICでプログラムを組むのと変わらないわけです。

回路もサブルーチン化

さて，BASICでプログラムを組むときも，あるまとまった処理をサブルーチンとしてブロック化してやると，プログラム全体がわかりやすくなります。もちろん，BASIC

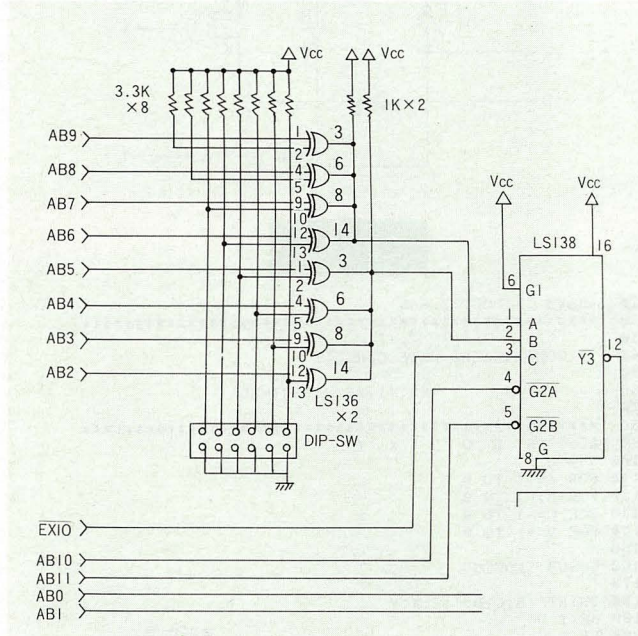
図３-a　MIDIのアドレスデコーダ部

図３-b　（参考）ワイヤードAND

リスト３

```
10 ' SAVE "1:LOGIC3.BAS
20 '****************************************************
30 '
40 '   ADRESS DECODER IN MIDI INTERFACE
50 '
60 '               63.11.23  A.SHIMADA
70 '
80 '****************************************************
85 PRINT TIME$
90 DIM AB(11)
100 DEF FNY(A,B,C,G1,G2A,G2B)=NOT(G1 AND (NOT G2A) AND
(NOT G2B) AND A AND B AND
(NOT C))
110 FOR ADR=0 TO &H7FFF
120 '
130 'ADR BIT
140 EXIO=(ADR >= &H1000)
150 ADR1=ADR AND &HFFF
160 FOR I=0 TO 11
170 AB(I)=-(ADR1 MOD 2)
180 ADR1=ADR1 ¥ 2
190 NEXT
200 '
210 GOSUB "LOGIC"
220 '
230 IF Y=0 THEN PRINT HEX$(ADR);" ENABLE!"
240 NEXT ADR
245 PRINT TIME$
250 END
260 '
270 LABEL "LOGIC"
280 P=AB(9) XOR -1
290 Q=AB(8) XOR -1
300 R=AB(7) XOR -1
310 S=AB(6) XOR -1
320 T=AB(5) XOR -1
330 U=AB(4) XOR -1
340 V=AB(3) XOR -1
350 W=AB(2) XOR -1
360 A=P AND Q AND R AND S
370 B=T AND U AND V AND W
380 G2A=AB(10) : G2B=AB(11)
390 Y=FNY(A,B,EXIO,-1,G2A,G2B)
400 RETURN
```

実行結果

```
RUN
18:13:37
0 ENABLE!
1 ENABLE!
2 ENABLE!
3 ENABLE!
Ok.
```

の命令自体も機械語からみればサブルーチンですが，それらを組み合わせて，より大きなサブルーチンにしていくこともできます。ICでも，デコーダやコンパレータのようにゲートを組み合わせた機能回路がひとまとまりで1個のICになっていたりします。これも大きなサブルーチンに相当するわけです。

たとえば，LS138というデコーダを，入力引数がA, B, C, G1, $\overline{G2A}$, $\overline{G2B}$, 出力が$\overline{Y0}$〜$\overline{Y7}$となっている定義関数に書き直してみましょう。すると，106ページの論理式1のようになります。

もっとわかりやすくサブルーチンの形にして，論理式2でもよいでしょう。

これを使って8月号に掲載されたMIDIボード用のアドレスレコーダ部をプログラミングしてみたのがリスト3です（図3-aに回路図）。

注意事項としては，LS136はXOR演算子をそのまま使ったことと，LS136の出力を4つつないでそのままLS138のA, Bに入力しているのは，4つの出力のANDをとっていることです（図3-b）。

LS138の出力は$\overline{Y3}$のみとっているので，定義関数FNYはA＝−1, B＝−1, C＝0のときのみ（もちろんG1＝−1,G2A＝G2B＝0）0となるようにしてあります。

このリスト3は，RUNさせると終了までたいへん待たされますが，結局イネーブル（Z80A SIOがON）となるのは，ADRが0〜3のときだけということがシミュレートされます。ところで，0〜3の区別はSIOの別の端子C/$\overline{D}$，B/$\overline{A}$で行っていますので，このアドレスデコーダ部では無関係でもかまいません。

応用として，D-FF（D型フリップフロップ）を使った4ビットカウンタのシミュレーションをリスト4に載せておきます。FFはクロックに従って動作しますが，クロックの立ち上がりでのみデータを保持するので，それを判断するのが少しややこしくなっています。サブルーチン“FF”は，データDとひとつ前のクロックCKとクロックCKXを入力とし，もしCK＝0かつCKX＝1（立ち上がりエッジ）ならばQ＝Dを出力し，それ以外はQを保存してRETURNするものです。残りの部分は皆さんでもすぐわかるでしょう。

もう，ハードはこわくない

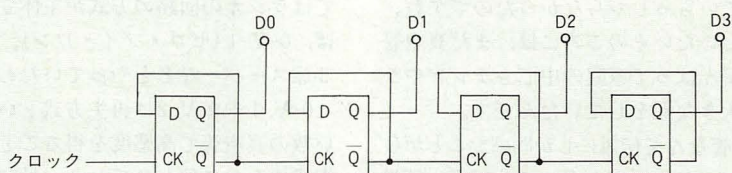
それでは，最初の問いに戻りましょう。なぜハードは難しく思えるのか，ということでした。それは，たとえて言うなら，BASICでプログラムを組むためにはBASICインタプリタのマシン語ルーチンまで理解しないといけないような錯覚に陥っているからではないでしょうか。

BASICは，命令の働きさえわかれば中身はそれほど必要ないのです。それと同じように各ICの働きを知りさえすれば，あとはICの足をつなぐだけで，とりあえずハードは理解できてしまうのです。リスト1からリスト4までのBASICプログラムが何をやっているのか理解できれば，デジタル回路そのものを理解したことになるのです。

確かに，実際の回路では電気を扱う上でのさまざまな処理が必要になってきます。しかし，それらはハードウェアが扱うデジタル情報を理解するうえでは特に問題になることはありません。

さあ，このコツさえ飲み込めば，次のステップはいろいろなICについて，どういう出力が得られるのかという入出力の関係に的を絞って勉強していくことになるわけです。いろいろなICの基本機能は三沢氏の記事でまとめられていますので，そちらを見てください。

そして，実際に工作の楽しさを味わってみたい場合には，ハード工作入門編を参考にしてください。要はICの足がハンダと導線でつながっていればよいのです。ではさっそく始めてみましょう。きっとパソコンに対する認識も変わってくると思います。

図4　4ビットカウンタ

リスト4

```
10 ' SAVE "1:LOGIC4.BAS
20 '***************************************************************************
30 '
40 '   4BIT COUNTER WITH D-FF
50 '
60 '                    63.11.23  A.SHIMADA
70 '
80 '***************************************************************************
90 '
100 CLK=0 : Q0=0 : Q1=0
110 '
120 IF CLK=-1 THEN PRINT CLK;D3;D2;D1;D0;-(D3*8+D2*4+D1*2+D0*1) ELSE PRINT CLK
130 CLKX=NOT CLK
140 Q=Q0 : D=NOT Q : CK=CLK : CKX=CLKX : GOSUB "FF"
150 QX0=Q
160 Q=Q1 : D=NOT Q : CK=NOT Q0 : CKX=NOT QX0 : GOSUB "FF"
170 QX1=Q
180 Q=Q2 : D=NOT Q : CK=NOT Q1 : CKX=NOT QX1 : GOSUB "FF"
190 QX2=Q
200 Q=Q3 : D=NOT Q : CK=NOT Q2 : CKX=NOT QX2 : GOSUB "FF"
210 QX3=Q
220 '
230 CLK=CLKX
240 Q0=QX0 : D0=Q0
250 Q1=QX1 : D1=Q1
260 Q2=QX2 : D2=Q2
270 Q3=QX3 : D3=Q3
280 GOTO 110
290 '
300 'FLIP-FLOP : INPUT D/CK/CKX
310 LABEL "FF"
320 IF CK=0 AND CKX=-1 THEN Q=D
330 RETURN
```

実行結果

```
RUN
 0
-1  0  0  0 -1  1
 0
-1  0  0 -1  0  2
 0
-1  0  0 -1 -1  3
 0
-1  0 -1  0  0  4
 0
-1  0 -1  0 -1  5
 0
-1  0 -1 -1  0  6
 0
-1  0 -1 -1 -1  7
 0
-1 -1  0  0  0  8
 0
-1 -1  0  0 -1  9
 0
-1 -1  0 -1  0  10
Break in 310
Ok.
```

●ハードウェア工作編

純粋なハード工作のすすめ

禁断の石の物語

Ookura Kenji
大倉 建二

ハードウェア特集の後半は楽しい工作の入門編。半導体技術の進歩によって高機能なデジタルICが回路の基本となった現在でも，真空管やトランジスタによるアナログ回路主流時代から受け継がれた自作の楽しみに変わりはありません。

伝説の石

業界の人々を含め皆さんが石，石と呼んでいるICですが，この元祖は，なにを隠すまでもなく，トランジスタなわけです。が，実はトランジスタよりもさらに前には真空管というものが使われていたのをご存じでしょうか。といってもせいぜい20年くらいのものですから，そんなに古い話ではないのですが，本誌の読者の方々の多くはまだ生まれていらっしゃらなかったのですね。まあ，だいたいそのころには，まだ真空管が結構がんばって家庭の中でもテレビやラジオで大きな顔をしていたんです。

「真空管なんて伝説にしか聞いたことがない」とおっしゃる方も多いでしょう。簡単に説明しておきますと，ガラスの筒がまずあるわけです。大きさはいろいろなんですが，だいたい茄子，そう野菜のなすびくらいのもの。それからキュウリくらいの太さで，長さが5センチから15センチくらいのものが多かったですね。そのガラス管の中に電極と呼ばれる，まあ鉄板ですね，それが組み合わさったのが置いてあって，その真ん中にはヒーターという電熱線がついているわけです。

もちろん真空管というくらいですから，密閉されていて中の空気は抜いてあって，ちょっと鉛筆の先で突くというようなことはできません。ガラスはよほど変わった種類のものでないかぎり透明なままですから中身はよく見えるのですが，多くは電極の一番外側のものが他の電極をくるむように作られていますので，電極の細かいところまではよく見えない。ところがヒーターだけはたいていニクロム線で作られていましたから，電気を入れると電熱器と同じような，赤っぽいオレンジ色というのか，そんな色で灯っているのが見えるわけです。

部屋を暗くすると，真空管のヒーターがみんな一斉に静かについているのが目で見えるんですね。それが単なる電源が入っているかを示す，パイロット・ランプではなく，真空管が動くために一番大切なことをしている姿だというのが大事なところです。電源を入れたまま裏ぶたをあけて，ああ働いているんだなあと，ぼんやりとしばらく見つめていたこともありました。

真空管とハード工作

真空管のラジオの場合，使っている真空管の数から4球とか5球とかいった呼び方がされていたのですが，ラジオ少年の世界ではラジオの回路の方式が主体で，たとえば，0-V-1(ゼロ・ブイ・ワン)，0-V-2，5球スーパーなどとやっていたわけです。

0-V-1や0-V-2は再生方式といい，少ない数の真空管で高感度を得ることができる方式のものに使われており，最初の数字が高周波増幅段の数を，最後の数字が低周波増幅段の数を示しています。たとえば，0-V-1でしたらイヤホンを鳴らすのが精一杯ですが，0-V-2になるとスピーカが鳴らせるようになるといった違いが出てきます。

スーパーというのは，スーパー・ヘテロダイン方式というラジオの回路方式の略称です。この方式は性能が非常に高くできるのが特徴でして，今のAMラジオはほとんどこの方式です。ただし，性能が高くなるかわり部品が多くなりますからキットも高くなる。そのようにして，キットにもずらっと値段の序列がつくわけです。

その後，主流がトランジスタになって大手電気メーカーも真空管を作るのをやめてしまうようになっても，しばらくは流通在庫がかなり大量にあったおかげで真空管の不足は起こらず，ラジオのキットも細々とは出回っていました。でもさすがに真空管のラジオ用部品自体が入手しづらくなり，いつのまにかそれらも消え去っていってしまいました。それでもしぶとく秋葉原などでは真空管の専門店が幾つもあるところをみると，いまだに真空管をあさって，なにか作っている人がいらっしゃるのでしょう。

オーディオ用のアンプでは真空管専用部品は真空管ソケットとトランスくらいのものですから，なんとか頑張れていて，保守部品としていまだに若干量は作られているアメリカやイギリスの軍用真空管がけっこう輸入されて使われているようです。残った国産品も需要がほとんどないのが幸いしてかいまだに健闘していて，なんと秋葉原のガード下では真空管のアンプのキットを扱い始めたところまで現れました。

「まだまだしばらくはなくならないよ」と心強い限りでした。

そういえば私も大学生のとき，粗大ごみのなかに真空管のラジオが捨ててあるのを見つけたものですから，すかさず真空管だけを抜いて帰り，家で短波ラジオを組み立てたことがあります。本当はまるごと拾っていきたかったんですが，さすがにそれを抱えて電車に乗り込む勇気はありませんでした。いま考えると惜しいことをしたと思いますね。

自作派の心

ところで真空管というのはヒーターが灯って，全体が温まってこないと電気的にはただの絶縁されて並んでいる鉄板に過ぎませんから，電源を入れても何十秒かはうんともすんともいわないんですね。電源を入れたあと一瞬の間があってブーンとトランスが鳴って，ヒーターが赤くなってきて，さらにじっとしていると，かすかに音ともつかないささやきが聞こえて，そしてザーという雑音のなかから目指す放送局からの音楽が聞こえてくるんですね。ここまでの間というのは，普通はただの待ち時間に過ぎないのでしょうが，自分でラジオを作ろうなどと考える者にとってはこの間（ま）がまた楽しいものなのですね。私も真空管が好きでして，家でも真空管のアンプを使っているのです。で，真空管とトランジスタはなにが違うのかと言われると，もちろん音の違いというものもありますが，やはりヒーターの存在と間というのが大きなものであると思うわけです。ところが，これが人には説明しにくいもので，じっと待た

されるのがよいといってもあまり理解されないんですね。

一度作ってみるとわかるのですが，たとえば友だちにラジオを見せたとしますね。2人でそれを囲んで，アンテナをつなぎ，電源を入れる。息をころしながら，じっと待っていると，じわーっと音が出てくるのと，電池が入っていてほらと手渡してパチンとスイッチを入れるといきなり大きな音がでてくるというのでは，やはり大きな違いがあると思うんですが，それが一度も作ったことのない人にはなかなか伝わらない。むきになって，いかに真空管のヒーターが美しいものか，じっと見ているのがいかに楽しいものであるかを説明してみてもせいぜい「マニアですね」とか「ロマンチストですね」で終わりですから，こちらとしても張り合いがない。何度か繰り返しているうちに結局，ニンマリ笑うだけにして終わりにしてしまうようになるんですね。真空管がトランジスタになり，ICとなっていくうちに，自作というのがだんだんと流行らなくなってしまったようです。トランジスタやICには真空管のヒーターから感じられたムードがないからじゃないかなんて思います。

もちろん，自作の衰退には，売っているものがあまりにも多機能，高性能でしかも安くなってしまったために，自分の手で作ることの意味が作るプロセスを楽しむだけで終わってしまうようになったことも挙げられるでしょう。妙に現実的ですが，これもまた大きな要因といえるでしょう。やはり作ったものに利点が少なくては面白くないわけです。買ったほうが安くついてしまうというだけでなく，なにも勝る部分がない，マネすらできないというのはつらいところです。

「自作」ということだけでもなんとか価値を認めてくれる仲間の場としては，いままでもそうですがアマチュア無線が最右翼でしたね。オーディオマニアもいるのですが，あちらはどうも高級志向というんですか，金に糸目はつけないというお金持ちの人が多くて，ただ自作したというだけですとあまり評価してもらえないような感じが私はしましたね。自作派のアマチュア無線家と称される人はだいたい，ジャンク屋をまわったりしながら，ごそごそと活動してたわけです。

デジタル回路とアナログ回路

さて，やや沈みがちだった自作派を活性化させたイベントが，マイコンの登場でした。最初に大きなブームとなったのが，かの有名なTK-80という日本電気が発売したワンボードマイコンです。内容的には，あくまでも評価用ボードとしての性格が強いものでして，どうひいきめに見てもとうてい「商品」と呼べるような代物ではなかったですね。ただ，それゆえ，アマチュアが登場する場があったということは評価しなくてはならないでしょう。

部品の絶対数も数えるほどしかなく，特殊な部品もありません。ですから，自作して評価用ボードを越えるものにすることも割と容易でしたし，しかも時代の先端である「コンピュータ」を自分の手にできるということで刺激された自作派は多かったはずです。

当時のマイコンに飛びついた人のなかに，かなりの割合でアマチュア無線家がいたということがそれを裏づけているように思います。アマチュア無線の世界でも，メーカーの無線機が主流となり，もう自作の時代ではないなどといわれて久しくなっていましたが，自作派は細々としかも，しっかりと息づいていたわけです。

ただ，マイコンの回路は，多くの自作派が長いあいだ親しんできたものとはかなりイメージの違うものでした。それまで馴染みだったラジオや無線機，あるいはオーディオアンプといったものがアナログ回路であったのに対し，マイコンはデジタル回路と呼ばれるものだったのです。

当時のアマチュア無線では大部分をトランジスタで組んでいましたが，高出力が必要な部分では真空管を使ったり，と思うとジャンクで見つけたICなども使ったりと結構バラエティに富んでいたんですね。それが，マイコンなどに使われるICとなると様子が少々違っているわけです。

マイコン，つまりデジタル回路の場合，回路図を見ても，いくつかのICが単に線でつながっているだけで，コンデンサはおろか高周波回路でお馴染みのコイルすらありません。抵抗も信号の流れのなかに入り込むことはほとんどなくて，まるでブロック図がそのまま回路図になっているように見えるのです。

私が初めてマイコンの回路図を目にしたときの第一印象は「プラモデルみたいだ」でした。やれインピーダンスだ，バイアスだ，回り込み，M結合だ，アースの引き回しだと，まあ部品をなだめすかしながら設計していくという側面がアナログ回路にはあるわけですが，それがデジタル回路になると何もないわけです。ほとんど何も考えずにつなぐだけで動くデジタル回路はまさに接着剤とシールだけで組み立てるプラモデルというイメージでしたね。自作派の無線家連中なら工具一式も持っていますし，TK-80程度の回路なら，口笛吹きながら作れてしまうわけです。

私もそうやって自分で回路図を引いて部品を買い集めて作った一人でしたが，作っていてもどうも電子回路らしくなくって，落ちつかないんですね。しかし，考えてみますと，デジタルICであるとはいっても，その中身はトランジスタの集まりであることに変わりないわけです。いくら「デジタル」といっても，電気というもの自体がアナログ量ですから完全にデジタルで動作するものであるはずはないわけです。

デジタルICとは

ちょっとむずかしそうな言葉を使いますと，デジタルICというのは結局，トランジスタの増幅器として使われる範囲外である，飽和領域と遮断領域を使っているわけです。遮断領域というのは，その名のとおり，トランジスタのコレクタ電流が流れない領域でスイッチでいうならOFF，飽和領域というのはベース電流をそれ以上にしてもコレクタ電流が増加できない領域でして，スイッチならON状態なわけです。

この両極端の状態を往復させて使うわけですが，どんなにクッキリと変化させようとしても，その中間ではどうしてもアナロ

グ的な動きが見られるわけです。理想的なデジタル回路があるとすれば，遅れ時間が0，出力が変化するときには，無限大の電流が流せ，しかも電線の抵抗も0ということになるのでしょうが，そんなことは絶対にないですね。

たとえば，一般的にTTLと呼ばれるICですが，これはひとつの出力に10個までつなげますと一般的にはいわれています。では，それなら11個なら絶対動かないのか，9個ならどんな条件でも大丈夫なのかといわれれば，そんなことはないわけです。カタログでの数値も，どの程度の負荷を与えたときに最悪でもこのくらいは保証しますとか，入力がこの電圧以上なら1とします，これ以下なら0としますといったことが決められているわけです。

要は，この条件なら1とします。0としますということを決めて，その範囲で動かすようにすればそれでデジタルICにできるわけです。あまり勝手な決め方をしていては周りが迷惑するだけです。

しかし一方，トランジスタの種別や回路方式によって作りやすいレベルというものもあるわけです。

だいたい，コンピュータでよく使われるものはTTLレベル，C-MOSレベルの2つが多くて，ほとんどはそれに準じています。特に高速を目指すようなところではECLと呼ばれるものが使われていて，一般的なものに比べて10倍から100倍くらいの高速動作ができるのですが，これがマイナスの電源が必要だったりと，特殊すぎるのが難点で，パソコンレベルで使われることはあまりないですね。

TTLからC-MOSへ

TTLとC-MOSの違いというのは，専門用語風にいうなら製造プロセスの違いということなんですが，要するに使っているトランジスタの動作原理が違うわけです。TTLの中で使われているトランジスタは，技術科の教科書に出てくるような，一般的なバイポーラトランジスタと呼ばれるもので，ベース電流をほんの少し変化させるとコレクタ電流が大きく変わるというものです。一方，C-MOSの場合では，FET，フィールド・エフェクト・トランジスタというもので，日本語では電界効果トランジスタなんて直訳したりするんですが，それを使っているわけです。

このFETというのは，電子の通り道に電界をかけて，その道幅を変えてやるといった方法で動きます。そのようなわけもあってか，バイポーラトランジスタのベースに相当する部分は，「ゲート」と呼ばれています。FETはゲートによって抵抗値を変えるように動くわけです。しかもこの制御には電界があればよいのですから理論的に電流が流れる必要はないんですね。このあたりの動作原理は真空管と似ていまして，実際に測定してもよく似た特性を示すのがまたFETの面白いところです。

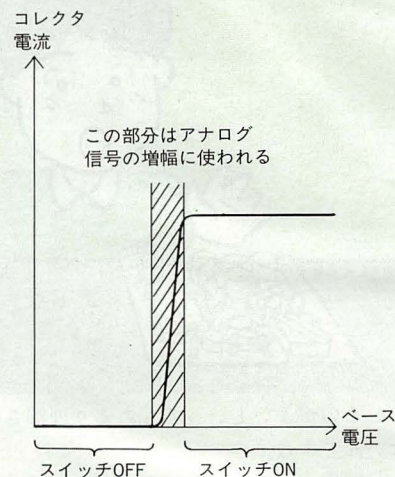



図1　トランジスタのスイッチング

さて，話を戻しますと，バイポーラ・トランジスタはいわば電流動作でして，どうあがいても，ベースやコレクタに電流を流

古きよき時代の工作少年

憧れの工作キット

じっと，真空管のヒーターの明かりを見つめているのが好きなラジオ少年やその予備軍の少年たちにとって，その明かりを自分の手で生み出すことができるラジオの組み立ては，憧れの的だったんですね。愛読していたのは『子供の科学』や『初歩のラジオ』。その最後の色のページにいつもあった神田の科学教材社の広告では，いろいろな模型などと混ざって，ラジオの部品一式を集めて，鉄板やアルミの箱に穴をあけたものと一緒にしたラジオのキットがたくさん載っているわけです。また，その売り文句が「キミにもできる」のような書き方ではなくて，「発売以来，5球スーパーとしては予想以上の感度であると好評をいただいております5球スーパーのキットです」といった具合なんですね。今にして思えば，ですます調の文句が，ちょっぴり大人扱いされているようで，そそられるものがあったわけです。

真空管自体も高価でしたから，たとえ1球ラジオ（つまり真空管を1本しか使っていないラジオ）であっても，小学生のお小遣い程度ではなかなか手が出せません。それでもクリスマスとお正月という2大イベントがあり，なんとかなりそうな現金が手に入ると，高嶺の花も花屋のかすみ草くらいにはなってくれるんですね。そして，ひょっとしてこのラジオ代を狙ってくるやつがいるんじゃないかなんて本気で心配しながら，しっかとカバンを押さえつけ秋葉原へと行くわけです。

ちょっと狭くて薄暗い店の中では，常連らしき人たちがぼそぼそと何かむずかしそうな話をしていたものです。そんななかで，ちょっと勇気を出して「〇〇のキットください」と言うわけですが，「ハイ」と言われてからキットの箱がくるまでの時間がすごく長く感じるんですね。そしてやっとキットが来て，手にすると不思議と軽いんです。

真空管の場合，家庭のAC100Vから数百ボルトという電圧を作るためにトランスが使われるわけですが，このトランスというのは銅と鉄の塊みたいなものですから，小型のものであってもかなり重たいものです。しかもシャーシと呼ばれるラジオの箱も，アルミよりも鉄板の場合が多かったですから，それがまた結構な重さがある。それらが一緒になってキットの箱のなかにあるのですから，実際には重たくて当然なのですが，浮かれたような状態で持ちますからあまり重いと感じなかったのでしょう。

電車に乗って箱をみると，その包み紙というのがデパートの包装紙のようなものではなくて，その店で売っている模型やらラジオやらの設計図の断片がちりばめてあるという，実にマニアックなものなんですね。その設計図をじっと見ていると，自分が買ったキットの回路図があったりするんですね。すると，また嬉しくて，その回路図で信号の流れを指で追ってみたり，選局用のバリコンのところなどは指で回すしぐさをしてみたりしてしまったりしましたね。

ハード工作は時空を曲げる？

家にたどりつくとこれがまた大変でして，しばらくは自分の寝る場所もないくらい，部品やらなんやらを部屋にちりばめて，本当に寝食を忘れて，とりつかれたように組み立てに入るんですね。母親のご飯ですよなんて言葉も，ほとんど雑音にしか聞こえない。なんで，人間は飯なんてくわなきゃいけないんだ，なんて真剣に思ったりしましたね。

組み立てに慣れていないのに，早く仕上げたい一心で説明書なんてろくに読みもしないで組み立て始めてしまうんです。おかげで余計に時間がかかったりもするのですが，組み立てている間はほとんど時間の経過というも

さないことには動きようがありませんでした。しかしFETは電圧動作で，出力は抵抗値が変化するようなものですから，それを受ける側が電圧動作をしてくれれば，電流などはほとんど流す必要がない，つまり低消費電力が期待できるわけです。

ICというのは，ちょっと考えただけでもおわかりのように，非常に細かく作ってあるために，熱が逃げにくいんですね。デジタルICは，ものを持ち上げるといったような物理的な仕事はせいぜいリード線を熱膨張させる程度のもので，ほとんどなにもしていませんから，消費電力のほとんどは熱に変わるわけです。おかげでちょっとトランジスタを詰め込んだりするとすぐに触っていられないくらい熱くなりそうになってしまう。ですから，低消費電力化というのは，集積度を上げるための大事なステップなわけです。その切り札としてC-MOSはとても有効な手段だったんですね。

C-MOSでは，少なくとも静的な状態ではまったくといってよいほど電流を流す必要がなくて，バイポーラよりも2桁から4桁以上小さく，しかもスイッチとして使ったときのON抵抗が非常に小さいということから，注目されたのですね。

ところが，このC-MOSというのもよいことずくめではなくて，電流こそ食わないのですが接合点に結構なコンデンサが形成されてしまうんです。FETの宿命ともいえるんですが，電界あるところにコンデンサ(キャパシタンス)ありというわけです。この形成されたコンデンサが，入力に並列に接続されたようになってきます。

先ほども述べましたとおり，デジタルICといえどもアナログから足を洗うことはできないわけでして，入力に並列にコンデンサがあれば，しっかりと積分回路が形成されて，0から1，1から0といった変化が入ってきてもそれがなまってしまうわけです。これが，動作の遅れに手を貸すことになってしまうために，たとえばTTLなら20nsくらいで0から1に変化してくれたのが，C-MOSのデジタルICでは100nsから200nsもかかるといったことになっていたわけです。

これを改善するには，全体を物理的に小さくしてしまうこと，すなわち微細加工技術によるほかないわけです。が，これがなかなか大変でして，一般的に使われるようになるにはしばらく時間がかかりました。

今ではC-MOSの速度も高速になりました。たとえば，今回のハードウェア工作編でも使われている74HCシリーズなどはまぎれもなくC-MOSですが，速度や出力などでは一般的なTTLである74LSシリーズに負けていません。そればかりか，Hレベル出力は圧倒的に大きくとれ，それでいて入力にはほとんど電流が流れませんから，ひとつの出力にたくさんのICをつないでもそれほど負荷にならないというわけで非常に優秀です。

カスタムLSIの代表であるゲートアレイなどでは，C-MOSでもゲート遅れが1ns以下なんていうのが当たり前の顔でのさばっていますし，68030や80386などの32ビットCPUではC-MOSが当然となっています。もう，C-MOSは遅いというのは当てはまらない世界です。

ところで，先ほども触れましたが，C-MOSのデジタルICの中心をなしているFETの動作は真空管とよく似ているわけです。真空管が，真空中を飛ぶ電子をコントロールしていたのに対して，FETでは半導体の中を通る電子をコントロールするわけです。いわば「固体真空管」とも呼べそうなFETがマイクロプロセッサ最先端で主流となっていることに，真空管の子供が活躍している姿を見るような感じをもつのは私だけでしょうか。

ともあれ，ハード工作の楽しみが，皆さんの石との結びつきとなって，次の世代の石の話を生むことになると，ひそかに期待しています。

のがわからなくなってしまうんですね。人間にとって時間というのはものごとの変化で捕らえられるものなんですね。好きな番組が始まる，終わる，日が昇り，沈み，星が出て，時計が回るわけで，それを取り払ってしまうと，時間というのは実際に停止するのではないかと思えます。

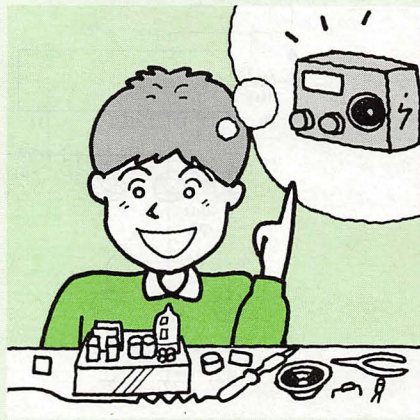
組み立てをしているときに確実なことは，それまで広告の写真でしかなかったものが，実際の部品として自分の目の前に現れ，それがしだいに写真のものと同じラジオとなっていく過程を一歩一歩と上っていくこと，それしかないですね。ところが，その動きは非常にゆっくりなわけです。シャーシをネジ止めして組み立てているときは素早く形になっていきますが，裏の部品のハンダづけは外観にはまったく影響せず，時間をかけている割にはあまり進んでいないように思える。時間の経過が遅くなったように感じるわけです。これが後半になるほど顕著になるんですね。つまり，組み立て始めてから段々と時間の経過が遅くなっていくように思えるわけで，これがラジオ組み立てにのめり込ませる魔力のひとつではないかとも思います。

感動の瞬間

最後の部品を取り付け，最後の1本の配線をすませると，全身の血が頭に上っていくようです。「組み上がったら，はやる気持ちを抑えて配線チェックをしましょう」などと説明書には書いてあるのですが，そんなことができるのは冷めてしまった大人だけ。実際にはとてもそんな余裕を持った心であるはずがありません。なんとか気を落ち着かせて配線チェックをしてみようとしても，深呼吸すら断続してしまうような状態ですから，回路図を追い掛ける手も震えてきてしまいます。

ヒューズをヒューズ・ホルダーに入れ，震える手でコンセントにプラグを差し込みます。電源スイッチに手をかけると，ガチンという音をシャーシに響かせて，あとは息もできず，真空管のヒーターだけをじっと見つめます。これまで見ていたヒーターと比べて，赤くなるのがずっと遅いような気がします。そして，じっと温まってきたころを見計らってボリュームを回していくとザラザラとかすかな音がします。そして，再生量調整をそろそろと回していくと雑音が急に増えていきます。再生方式では高感度を得るために真空管を発振すれすれで使うため，あるところまでいくとギャーと発振するのですが，最初はその感触がつかめず，何度かギャーギャーととんでもなく大きい音を立てることになります。そして感触をつかむと，こんどは選局用のバリコンを回していきます。すると，途中で「ひょ！」となにか聞こえます。そろっと戻すと――この時間がまたとてつもなく長く感じるのですけれど――人の声が聞こえてきます。何を話しているかなど関係ありません。すべてが神の声のように聞こえてきます。この瞬間，張り詰めた気持ちは吹き飛ばされ，全身の力が抜け落ち，頭に上っていた血の全てはなくなり，空っぽになった，虚脱感だけの世界です。そのうち，親から「うるさい！　何時だと思ってんの！」のひと言で我にかえるわけです。

こうして作ったラジオは，自身の満足はもちろんのことですし，見かけは悪くても自作となれば友だちも一目おいてくれました。回路図を前にして，かじりかけの専門用語を並べ，やや気取った会話をする同じ趣味の者同士のなかでも，ちょっとよい気分が味わえたのを覚えています。

●ハードウェア工作編

初歩からの電子工作

電子サイコロを作ろう

Suzuki Norio
鈴木 典雄

それではデジタル回路の実践編に入りましょう。まずは簡単なところから，ICを2個使って発振回路とカウンタで電子サイコロを作ります。アルゴリズムに従ってゲートを組み合わせていく過程をじっくりとご覧ください。

今年もMOS ICには辛い季節がやってきました。乾燥した部屋で金属の部分に触れるとピリッと衝撃が走ります。そう，こいつがMOS ICの天敵，静電気です。LSIは熱に弱いとかMOS ICは静電気に弱いといった通説がありますが，最近のMOS は保護回路が入っていたりと結構丈夫にできていますので怖がらずに電子工作にチャレンジしてみてください。ここでは工作の入門として，電子サイコロを作ってみることにします。先月号のROGUEスゴロクにでも使ってください。

さて，デジタル回路でサイコロを作るというと数字で表示する，というものになりやすいのですが，それではあまりに能がないのでLED（発光ダイオード）でサイコロの目を表示させてみましょう。ついでに目が変わるごとに“ピッピッピッ”と音を出すようにしましょう。

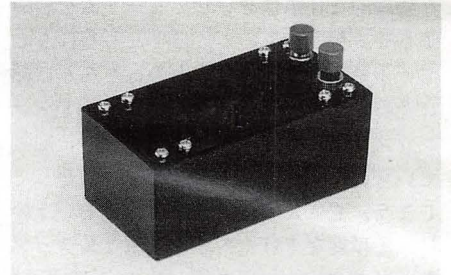
IC 2個でできますので部品代もたいしてかかりませんし，特殊なパーツも使用しません。これからハード工作の入門をしていこうという方でも簡単に作れると思います。ハードは初めてという方もとりあえず，これを機会に工作に慣れておいてください。

まずは回路説明

回路を設計していくのはプログラミングの過程にも似ており，工作自体はプラモデルを作るようなものです。デジタル回路は配線を間違えなければたいてい動いてしまいますが，どうして動いているのかを理解せず工作だけをやっていても成長はありません。できるだけ詳しく解説しますのでじっくりと回路図を追ってみてください。

図1　全回路図

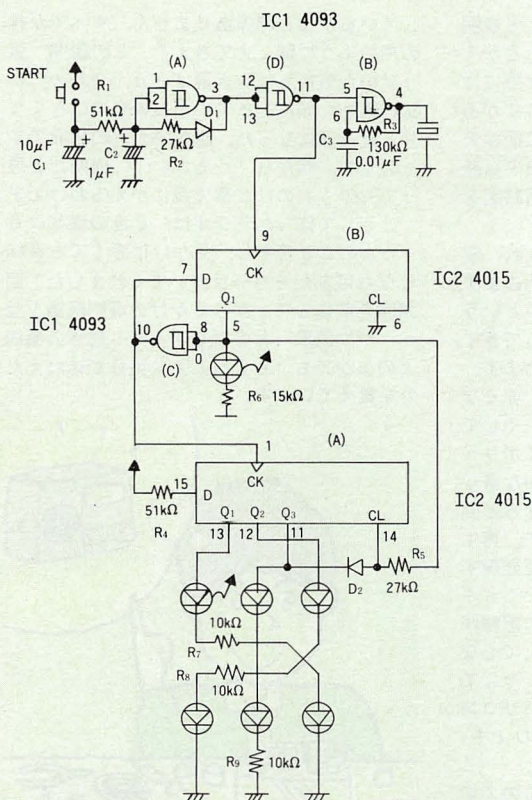
まずは動作原理から。サイコロというと乱数の発生回路が必要になるように思われますが，実際のサイコロを見てもサイコロ自体にはなんの細工もなく乱数的な要素は含まれていません。サイコロを投げる人間の動きに乱数の要素が含まれているのです。そこで人間の行動を乱数発生器として考えれば，サイコロの動作は非常に簡単なものですみます。

具体的に考えてみましょう。サイコロは順繰りに数字の目を表示するだけでも，人間がスイッチを操作するタイミングによって目を決定すれば十

分にランダムな値が得られそうですね。必要なのは次々に目を変えること，そして必要なときにそれを止めることとなります。

シュミットトリガによる発振回路

サイコロの目を動かす原動力となるのは一定の周期で信号のON/OFFを繰り返す発振回路と呼ばれるものです。デジタル的な発振回路にはふつうのインバータを使ったマルチバイブレータとシュミットトリガゲートを用いたものがありますが，ここではゲート数が少ない後者を使います。すでに「マルチバイブレータとはなんぞや」,「しゅみっととりがげーとってなんれしゅか」という方もいると思います。マルチバイブレータはおいといて，シュミットトリガゲートだけ説明しましょう。

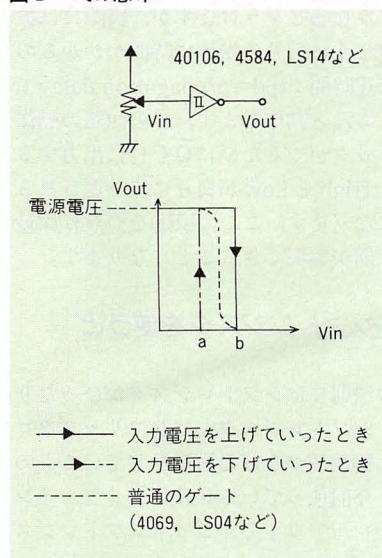
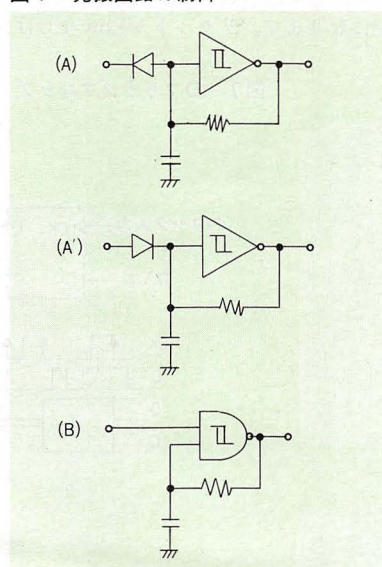
シュミットトリガゲートというのはヒステリシス特性を持ったゲートのことです。デジタル回路では電圧や電流の量がある一定の値を越えているかどうかで0と1の2つの状態を表しますが，ヒステリシス特性というのは，たとえば図2のような回路を組んでボリュームで0Vから電圧をあげていったときと，最大電圧から徐々に弱くしていったときとで状態変化する点が異なるということを意味します。こういった特性を持つゲートはゲートの中にヒステリシス記号を書いて表現されます。

図3を見てください。このゲートの入力端子にはコンデンサと抵抗がつながっています。コンデンサは放電した状態なので両端の電圧は 0V です。このゲートの入力はコンデンサの両端の電圧となりますので，

当然ゲートの入力はLowとなり，このゲートは（インバータなので）Highの出力を出すことになります。この出力は再び抵抗を通して入力に戻っていますね。この電流は少しずつコンデンサに充電され，コンデンサ両端の電圧は上昇していきます。そして電圧がa点を越えたとき，ゲートの出力は反転し今度は出力端子に向かって放電が始まります。そしてb点までくるとまた反転し，先ほどの状態に戻ります。以後はこれをずっと繰り返し，このゲートから外部への出力は見事に発振しているというわけです。

次にこの発振を制御することを考えてみましょう。図4(A)は図3にダイオードをぶら下げたものです。(A)でダイオードの左側をHighにしてみましょう。左からの電流はゲートの入力へ流れようとしますが，一方通行のダイオードでせき止められ，結局ゲートは図3のときと同じ動作をしますね。今度はダイオードの左端をLowにします。するとダイオードの右端のほうの電位が高くなって電流の向きが変わり，ゲートへの入力はみんな外へ逃げてしまいます。よって発振は停止します。(A')はダイオードの向きを逆にしたもので，これとは発振の条件が逆転しています。理由については自分で考えてみてください。

いまの説明では強引にダイオードで発振を止めていましたが，(B)はもう少しスマートです。インバータの代わりにヒステリシス特性を持ったNANDゲートを用い，ひとつの端子を制御入力として使っています。NANDゲートの真理表を思い出してください。ここでひとつの入力を High に固定してやるとこのゲートはインバータと同じ働きをします。また，Lowに固定すると，もう一方がどんな電位であろうとゲートの出力はHighのまま変わりません。つまり，制御入力のHigh/Lowによって発振が制御できるのです。

これで発振自体の制御はできましたが，さらに発振の周波数を制御することを考えてみましょう。一般にこのような発振回路は抵抗とコンデンサのバランスで周波数が決定されています。たとえばコンデンサの容量を大きくしたり，抵抗の値を大きくするとコンデンサの充電は遅くなり周波数が下がっていくわけです。要はコンデンサの充/放電速度によって周波数が決まっているのですから，なんらかの原因でゲートの入力電圧が変化すると周波数を変えることができます。

図5を見てください。ここではゲートからの出力はダイオードでせき止められ，入力側には回ってきません。ではどうやってコンデンサを充電するのかというと，Vin という端子から電流をもらっています。つまりVinの電圧で周波数が制御できるわけです。ゲートの入力端子の電圧が上がって出力がLowになるとコンデンサに溜まった電気はダイオードを通して一気に放電します。したがってこの回路の出力は図のようにLowの期間が非常に短くなります。また，Vinの電圧が下がるとやがては発振も停止します。

ここで図1を見てみましょう。IC1の(B)部分はもうわかりましたね。この出力はブザーにつながっています。(A)の回路は図5と同じ機能のものですがLowの期間を長くするため抵抗が入れてあります。スタートスイッチを押すとコンデンサ C1 に充電され，発振が始まります。スイッチを離してもコンデンサの放電は続き，発振は数秒間継続しますが，コンデンサからの電圧は徐々に下がっていきますので発振周波数はだんだん低下し，ついには停止します。なお，(D)の回路は発振が終わったときブザーも停止するように入れてあります。

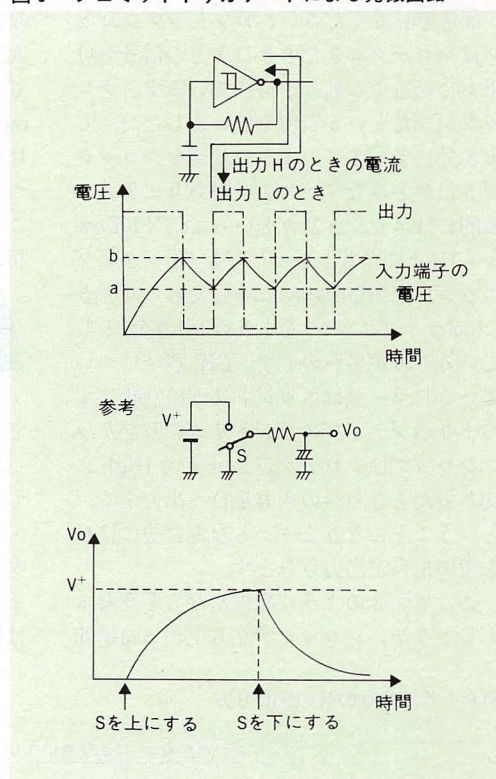
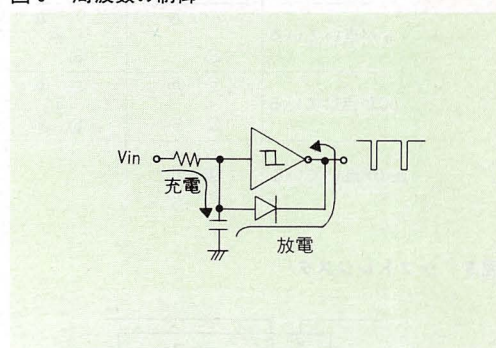



図2　⊏̲の意味

図4　発振回路の制御

図3　シュミットトリガゲートによる発振回路

図5　周波数の制御

目の法則を考える

発振ができたら次は目の表示です。図6をじっくり見ていると2つの法則に気づくと思います。まず，真ん中のLEDは数字がひとつ進むたびに点いたり消えたりを繰り返しています。もうひとつ，まわりのLEDは必ず2つずつの組みになって点滅します。こういったLEDの動作を実現するのがDフリップフロップであり，シフトレジスタな

のです。

図７を見てください。ＤフリップフロップはクロックがきたときＤ入力の信号をＱ出力に転送する働きを持っています。クロックがきたという表現を使いましたが，ICの番号によってどのような状態をクロックがきたかとみなすかに違いがあります。一般的にはエッジトリガといって信号がLowからHighに立ち上がったとき（ポジティブエッジ）やHighからLowに変わったとき（ネガティブエッジ）をトリガ（引き金）としているものが多いのです。CK端子についている▷マークはエッジトリガの意味です。つまりポジティブエッジトリガのＤフリップフロップはクロックがLowからHighに変わったときＤへの入力をＱへ出力する，ということになります。ちなみに$\overline{Q}$にはＱと逆の信号が出力されます。

ここで図８のように配線したらどうなるでしょうか。クロックが立ち上がった瞬間にＤ入力が変化してしまうのでうまく動かないような感じをうけますが，実際には拡大図のように信号の伝達に時間がかかるので（伝達時間：tpd＝propagation delay time）きちんと動作します。この回路の動作はクロックがくるたびに$\overline{Q}$をＱに出力する，つまりHighとLowが交互に繰り返されることになります。ここにLEDをつなげば最初の法則が実現できることになります。

シフトレジスタを使うと

次の法則にはシフトレジスタがぴったりです。シフトレジスタとはＤフリップフロップを複数個つなげた構造をしているものです。今回使っている4015はシリアルイン・パラレルアウトの４ビットシフトレジスタと呼ばれるものです。図9を見てください。まず，リセットがHighならばすべての出力はLowになります。リセットがLowならば，次のクロックでＤの入力がQ_1に転送され，また次のクロックでそれがQ_2へとどんどん次のフリップフロップにつながっていきます。それと同時にまた新しいＤ入力が現れます。ちょっと考えると，Q_1からQ_4まで全部Ｄ入力が出てしまいそうですが，先ほどのtpdのおかげでそうはなりません。

このようにひとつ入力があるたび出力がある方向に移っていく（シフトする）のでシフトレジスタというのです。また，レジスタですから，なにもなければずーっと同じ状態のままでいます。さて，ここでＤ入力をHighに固定してやるとどうなるでしょうか。もうおわかりでしょう。クロックが入るたびにHighである出力が増えていきます。そしてクロックをさっきの$\overline{Q}$からとれば真ん中のLEDが消えたとき，ひとつずつ進むことになります。これで目的は達成されました。

ただし，このままでは６の目が出たあと

図６　サイコロの目の変わり方

	aが点かない	aが点いている
		ⓐ
bが点いている	ⓑ ⓑ	ⓑ ⓐ ⓑ
bCが点いている	ⓒ ⓑ ⓑ ⓒ	ⓒ ⓑ ⓐ ⓑ ⓒ
bcdが点いている	ⓒ ⓑ ⓓ ⓓ ⓑ ⓒ	

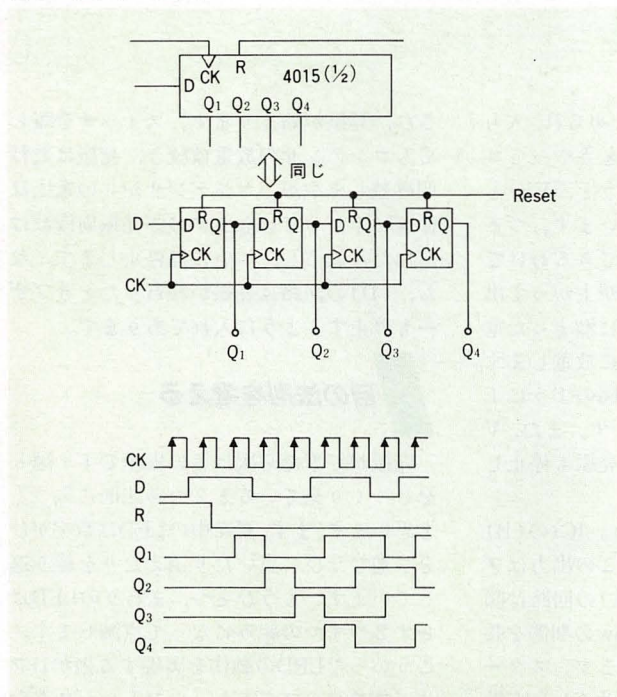



図７　Ｄフリップフロップ

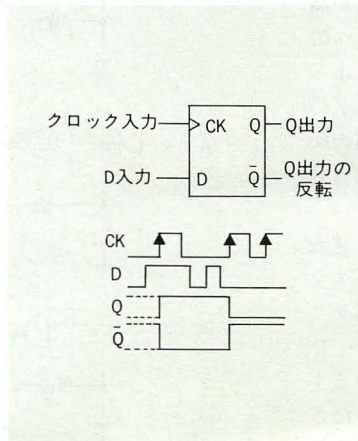



図８　Ｄと$\overline{Q}$をつなぐ

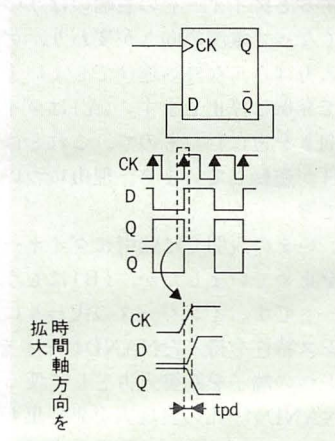



図９　シフトレジスタ

図10　リセット回路

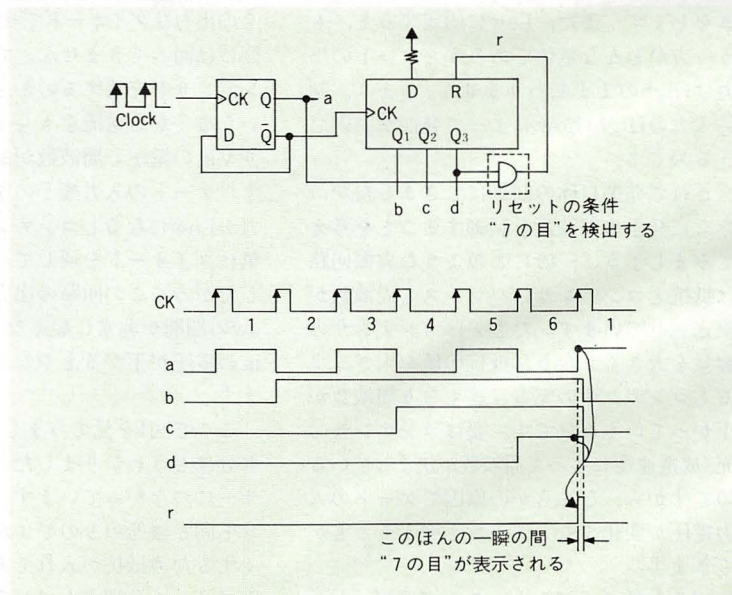



図11　シフトレジスタをフリップフロップにする

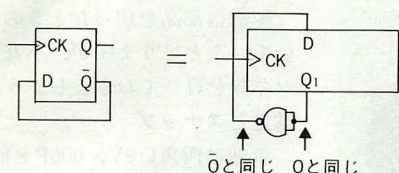
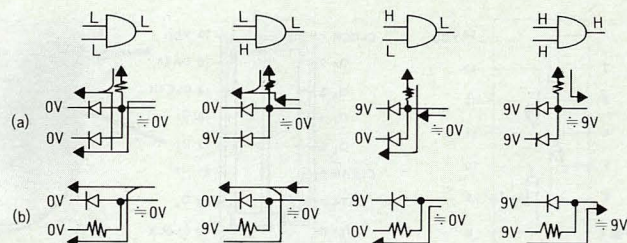



図12　ダイオードと抵抗によるANDゲート

は7の目が出てしまいますから，これを防止する回路を加えます。どんなとき7になるかというと，それはa，b，c，dすべてがHighのときですが，回路が定常状態の場合，dがHighならb，cは絶対にHighとなっていますから，結局aとdがHighのときが7の目ということになります。したがってaとdのANDをとってリセットに加えるという論理回路をつけ加えてやります。

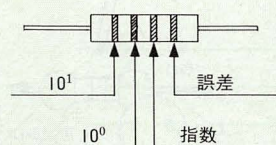
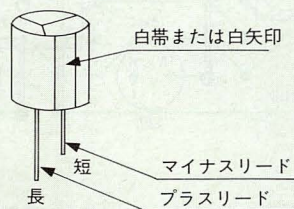
では7の目で止まってしまったらどうなるのでしょうか。このときはリセットがかかり，結局1の目になって一件落着するわけです。

これで必要な回路は揃いました。回路図を読んで流れを追ってみてください。注意点としては，4015という石には2個のシフトレジスタが入っているのですがシフトレジスタはひとつしか使いません。そこで余ったもうひとつのシフトレジスタをフリップフロップとして使っています（もともとフリップフロップでできているのですから）。問題は$\bar{Q}$の出力がないということですが，なければ作ってやればよいのです。ふつう$\bar{Q}$にはQを引っくり返した値が出ていますから，インバータでQの値を引っくり返してやります。図11のような回路を組んだとき定常状態では$\overline{Q_1}=Q_2$となっていますが，だからといってQ_2を$\overline{Q_1}$の代わりに使ってDにつないではいけません。なぜでしょう。考えてみてください。

もうひとつ，回路図にはANDゲートが見当たりません。これは図12のように抵抗とダイオードで置き換えられているからです。図に各入力がHigh/Lowのときの電流の動きを書き入れてありますので，皆さんで回路を読んでみてください。

表1　部品リスト（参考価格）

パーツリスト	個数	単価(円)
IC：4015	1	95
4093	1	55
抵抗：P型カーボン		10
10kΩ	3	
15kΩ	1	
27kΩ	2	
51kΩ	2	
130kΩ	1	
コンデンサ		
電解　1μF	1	15
10μF	1	15
マイラー　0.01μF	1	15
セラミック　0.01μF	1	15
LED　赤色丸型	7	25
ダイオード　1S1588	2	20
電池スナップ	1	30
電池　006P 9V	1	180
スイッチ：プッシュON	1	100
オルタネート	1	200
基板　ICB-88	1	90
ケース	1	300
アクリル板	1	300
圧電素子	1	50
スズメッキ線	少々	
配線材	少々	
スペーサー　15mm～20mm	4	20

図13　部品の見分け方

カラー	数値
黒	0
茶	1
赤	2
柿	3
黄	4
緑	5
青	6
紫	7
灰	8
白	9
金	5%
銀	10%

製作編―まず部品から

お待たせしました。それでは製作に取りかかりましょう。まずは部品集めからです。特殊なものはないので近くにパーツ屋さんがあればだいたい揃います。近くにパーツ屋さんがなければ通信販売を利用するという手もあります。秋葉原の秋月電子，ヒロセ無線などは通販もやっているはずですから『トラ技』の広告などを参照して問い合わせてみてください。

IC

4015，4093ともにどこのメーカーのものでもかまいません。たとえば，モトローラならMC14015B，日立ならHD14015B，東芝はTC4015Bというぐあいに型番に4015という数字が入っていますので，お店の人に“しいもすのよんまるいちご（C-MOSの4015）”といえば通じるはずです。

コンデンサ

電解コンデンサは耐圧16V以上ならOKですが，あまり大きいと形も大きく値段もはるのでなるべく小さいほうがよいでしょう。

抵抗

1/4WのP型カーボン被膜抵抗器です。1本5円くらいからありますが，まとめて買うと100本100円とかで売っていることもありますので，電子工作をやっていこうという人はよく使いそうなものはまとめて買っておくとよいでしょう。

LED

なんでもいいのですが，サイコロの目はふつう丸いので丸型のものを買ってください。あまり大きいとICとぶつかりますので

図14　端子表

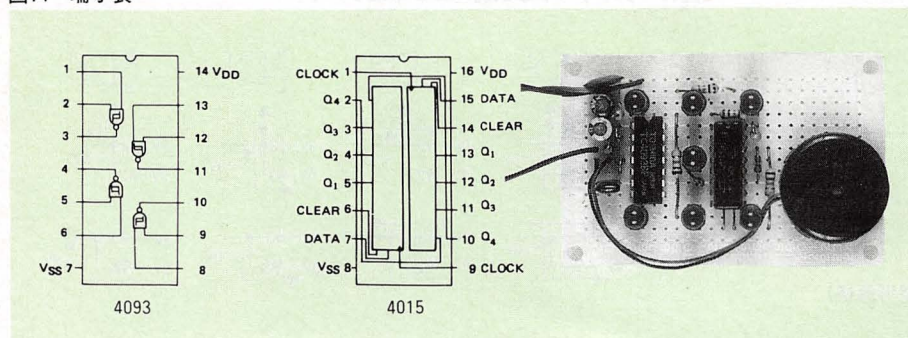
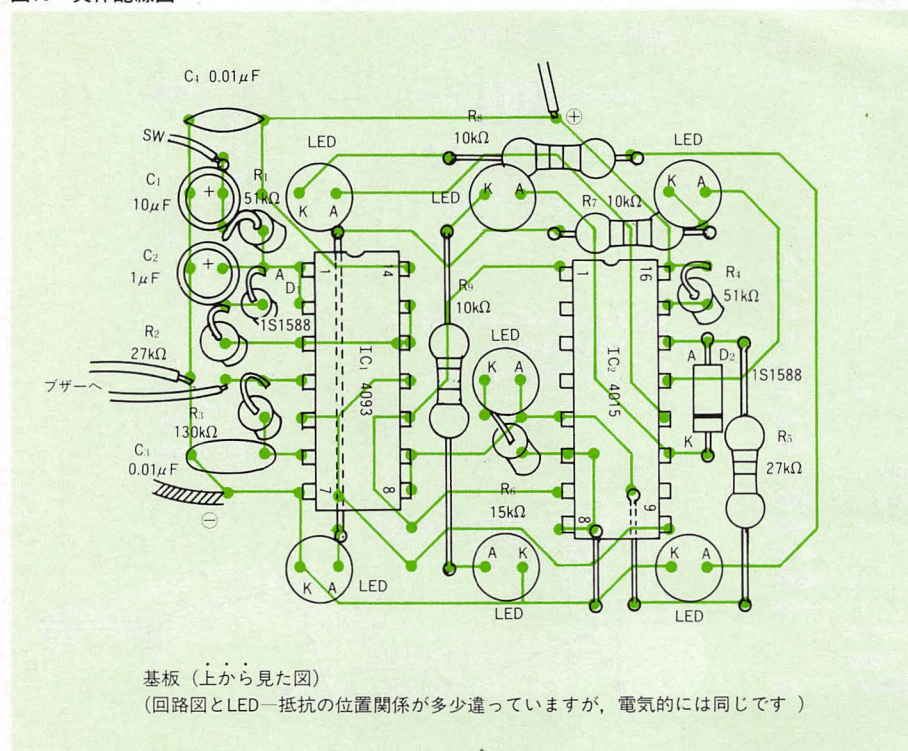



図15　実体配線図

GL5PR5くらいまでのものを揃えてください。

ダイオード

シリコンのスイッチングダイオードです。

圧電素子

ブザー用のもので発振回路を内蔵していないものを選びます。薄っぺらくて丸い形をしています。

ICソケット

なくても大丈夫ですが，ハードに自信のない人は必ず使ってください。できるだけ背の低いものを選びます。

基板

サンハヤトのICB-88を使いましたが，ほかのものでもかまいません。ICピッチのもので15×20目あれば十分です。

スイッチ

プッシュスイッチは押すとONになるものです。世の中ひねくれものもあり，押すとOFFになるタイプもありますので注意します。あと，オルタネートといって一度押すとON，もう一度押すとOFFになるものもありますがそれも避けたほうがよいでしょう。電源スイッチはトグルでもスライドでもなんでもいいです。

ケース

基板が入り，なお少し余裕があるものがいいでしょう。プラスティックのケースを使い，ふたはアクリル板に取り換えます。アクリル板はスモークドブラウン（茶色半透明）かスモーク（黒半透明）がよいでしょう。

スペーサー

基板をケースに固定するためのものでネジつきのものを使います。LEDの足をよく見てください。出っぱりがあるはずです。LEDはそれ以上基板に入らないことになっていますから，それを考慮してスペーサーの高さを決めてください。

スズメッキ線

配線は部品を切ったときの余りの足でたいていこと足りますが，不安な人はスズメッキ線を買っておきましょう。

電池とスナップ

電池は四角い9V，006Pを使います。電池スナップは006Pの電極につける金具(?)です。

いよいよ組み立て

部品が集まったら組み立てにかかります。まず，部品を取りつける順序です。プリント基板の場合は背の低いものからというのがふつうですが，今回のような穴あき基板を用いる場合，ICソケットなどからつけていくと位置決めが楽になります。が，今回の基板の場合ICソケットの下にジャンパ線が3本もあるのでICソケットより先にジャンパを配線しないとあとでえらい目にあいます。

ということでジャンパ線をつけたらICソケットをつけていきましょう。そうすると部品の配置が決まるので，あとは端から順序よくつけていってください。部品を基板に差したら裏側で部品の足を配線図どおりに曲げ，次の部品の足の位置で線を切ります。次の部品も同様に足を折り曲げ，いま差した部品の足下をハンダづけしていきます。このようにするとわりと綺麗にしあがります。ただし，LEDの足は折り曲げるとLEDが傾きますので曲げずに切ったほうがよいでしょう。また，配線のすんだところは実体配線図に赤鉛筆で塗りつぶしてチェックするとよいでしょう。

電解コンデンサやLED，ダイオードは極性が決まっていますので注意してください。LEDと電解コンデンサは足の長いほうがアノード(＋)，ダイオードは1S1588の場合，青い帯のあるほうがカソード（－）になります。ICは表から見て切り欠きの左下が1番ピンです。切り欠きがないときは1番ピンの近くにポッチがあるのでわかるでしょう。ICソケットを使う人はICは最後に取りつけます。

配線が足だけでは足りなくなったときは，いままでに切って余った足を継ぎ足すか，スズメッキ線を使って配線します。スズメッキ線などは，配線前にハンダメッキをしておくと作業がはかどります。それから，スズメッキ線を長く引きずるときは曲がり角でポツポツとハンダづけして固定してください。

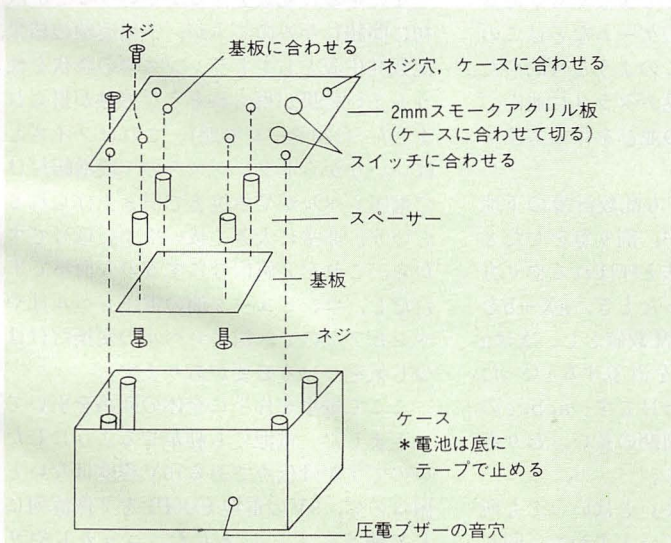
基板のチェック

基板ができあがったら，チェック作業に移ります。基板の裏を見てください。イモハンダやショートしているところはありませんか？　配線の間違いや，やり残しはありませんか？　それがすんだら図16の要領で回路をチェックします。まず，(a)の要領でショートのチェックです。電池のマイナス側だけをスナップにつなぎ，テスタを電流計にして回路に流れる電流を測定してみます。針が振れなければ正常ですが，振れた人はショートしている場所を探してください。

次はICの電源のチェックです。今度は電池をちゃんとつなぎ，テスタを電圧計にしてICソケットの7または8番ピンに－，14もしくは16ピンに＋をあててみます。電池と同じ電圧を示せば正常です。

ここまで大丈夫なら電池をはずし，ICをソケットに入れてみてください。そして(a)と同様のテストを行います。LEDの点き方によっても違いますが，0～3mA程度が正常です。電池の＋とR1のリード線をショートさせてみましょう。LEDが点滅しブザーが鳴りますね。指を離してそれがだんだんゆっくりになって止まればできあがりです。

チェック項目が一応正常でなお，動かないという人は以下の部分に重点を置いて再チェックしてください。

・圧電ブザーが鳴らない：IC1の左側
・点かないLEDがある：LEDの配線，極性
・真ん中が光るだけ：IC1の下とIC2との間
・7と6を繰り返す：IC2の右側

そして完成へ

あとは完成した基板をケースに組み込んでしまいます。ケースにはたぶんアルミのふたがついていると思いますが，これをアクリル板に換えてしまいます。ふたと同じ大きさにアクリル板を切り，ネジ穴を開けます。次にそのアクリル板に基板を取りつけるための穴とスイッチ用の穴をあけます。そうしたらスイッチは取りつけてしまって結構です。レコードスプレーがある人はケースとアクリル板に吹きつけておいてください(なければ省略)。これは静電気防止のためです。

基板につけた電池スナップの＋側をはずし，基板の＋のところとSWのところに予備配線を行います（あとでスイッチにつなぐ)。電池と圧電素子をテープでケースに固定し，基板を入れスイッチを接続します。あとはスペーサーで基板を固定してふたをすればできあがりです。

ご苦労さまでした。初めてハードを作ったという方はどんな感想を持たれたでしょうか。自分の手で作ったものが動くというのはなかなか感慨深いものがあるのではないでしょうか。この気持ちを忘れずに，もっとほかの工作にも挑戦してみてください。

図16　チェックのしかた

図17　組み立て方

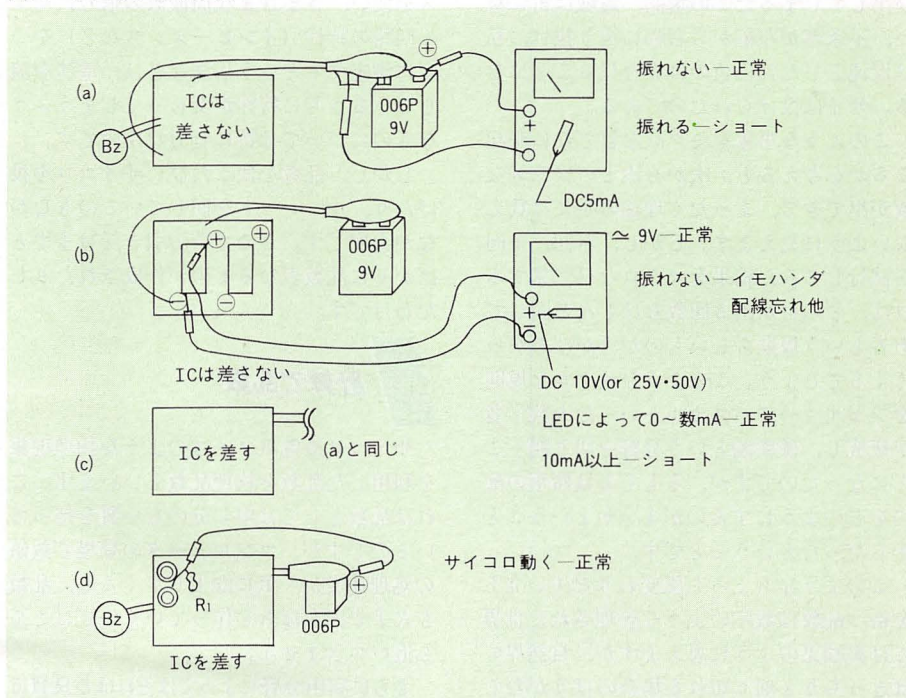
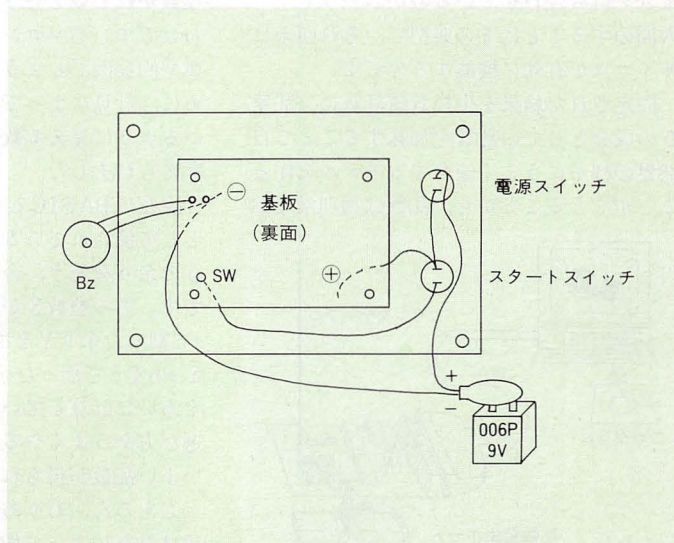



図18　スイッチの配線

●ハードウェア工作編

実録　乱数発生器の設計と製作

大きなノイズの使い方

Kuwano Masahiko
桒野 雅彦

ハードの人，桒野氏は半導体が発生するノイズを逆利用して乱数発生器を作ったようです。いろいろな微調整を加えながらデジタル回路とアナログ回路を組み合わせていきます。その設計過程を中心にハードウェアの設計法を追ってみましょう。

その瞬間，立方体は私の手による制御を離れ，あと戻りすることの許されない空間を浮遊し，やがてテーブルの上を跳ね厳然とひとつの数字を表すことになる。人知によってたどりつくことのできない世界を呼び出してしまったこの瞬間，論理は宙に舞い，不条理が左脳から右脳に突き抜け，私は混沌とした乳白色の世界を見ることになる。サイは投げられたのである。

このような事象をミクロな目でなにが起こるかを考えると，次から次とデタラメな数が出てきて，まったく理論の手には負えないように見えます。しかし，百回，千回と試行してみた結果をマクロな目で見てみれば，どの目が出る回数もほとんど同じであるという規則らしいものが浮かびあがってくるでしょう。このようなマクロな規則をフェルマーやパスカルといった大数学者が研究し，確率論という分野を切り開くことになったのですが，そもそもは賭事の配当をどのようにするのがよいかということからだったということです。

このようなちょっと風変わりな生い立ちを持つ乱数は数学のような管理された世界では異端児のように思えますが，自然界ではきっちりと割り切れる場合のほうがむしろ珍しく，なにごとにつけてある程度の乱雑さを持ち合わせているのがふつうです。人間のすることはその典型で，それゆえにサイコロが有効に機能するのです。

固定された結果を生む数値計算に，乱雑さの成分としての乱数を加味することで自然界の様子にうまく適合するモデルを作ることができることから，乱数は物理学を始めとする各種のシミュレーションにと利用されるようになりました。正確な値よりも，全体的な傾向を知るために乱数で適当に揺さぶってみるというのは簡単でしかも便利な方法です。乱数の電気信号版であるノイズを使い，さまざまな周波数の信号に対する回路の特性（インピーダンスなど）を一度に測定するという手法などは，電流を流しているうちに特性が変わってしまうようなものについては特に有力な手段です。

しかし，乱数を使うたびにサイコロを投げたり，ルーレットを回しているのもなかなか大変です。そこで苦労引き受けますとばかりに乱数表なるものが作成されたりしたわけです。

計算で乱数

サイコロや電気ノイズのような物理現象を利用した乱数を物理乱数といいます。これは乱数としては申し分のない質を持っているのですが，コンピュータの登場で数値の処理速度が一気に向上してくると，乱数も人手でぽちぽちと作っていくのではとても追いつきません。

さらに利用分野によってはそれほど良質でなくてもいいから，とにかくデタラメそうな数がほしいということも少なからずあるわけです。パソコン上のゲームなどはこの典型的な例でしょう。このような目的のために，計算によって一見デタラメに並んでいるように見える数字の並びを作る方法が考えられました。

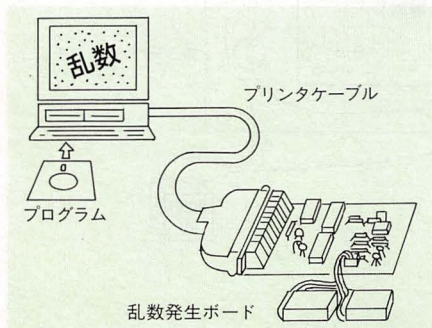
現在のBASICなどでも乱数計算の下地として使われているのが，割り算をしたときの余りを使う，合同法と呼ばれるやり方です。ある整数Xがあったとき，aX＋bをcで割った余りYを次の乱数値とし，次はaY＋bをcで割った余りを計算するといったぐあいに計算していくわけです。a, b, cの選び方をうまくやると周期の長い，わりあいよい乱数が得られます。

もちろん，わりあいよいとはいっても所詮は計算によって作っていますから，同じ数が2度連続して出てくることがないなど，真の乱数として見るには欠点が少なくありません。2度同じ数字が出てくるのは，適当に割り算をして下位の桁を落としてしまえばなんとかなりますが，根本的にはどうにも対策のとりようがない面があります。このような数値並びがあくまで疑似乱数としか呼ばれないのはこのためです。

特に学術用に使われるときには乱数の質が問題になりますから，どうしても疑似乱数では困ることになります。ある程度大きなコンピュータでは乱数発生器というハードウェアが用意されていて，物理乱数を得られるようにしています。しかし，パソコン用としては圧倒的な種類の周辺ボードを擁するPC-9801にしても，乱数ボードなるものが売られているのを見たことはありません（みんな疑似乱数で我慢しているのだろうか)。雑誌でも取りあげられたことはほとんどないようで，少なくとも私の手元の本のどれにも物理乱数の発生などというものは取りあげられていません。今回は，この隙間をついて乱数発生器を作ってみることにしましょう。

乱数発生器の設計

なにを基に乱数を作るかということが最初に問題になるのですが，予備実験の結果，乱数発生源としてトランジスタの降伏を使うとよい結果が得られそうな感触が得られました（120ページ参照)。これはノイズとはいいながらトランジスタの1段増幅だけで電圧レベルがV単位まで引き上げられるという，非常に大きく扱いやすい信号ですから，これを2値信号にするのも簡単です。ただし，コンピュータ側の電圧レベルはややシビアなので，信号レベルの変換だけは少し気をつける必要があります。

ここで電卓を片手に全体の回路を引いていきました。電池でも動かせるようにしたいのですが降伏点である10V程度はないと困るので，9Vの電池（006P）を2個直列にして使うことにしました。ニッカドやリ

チウム電池と違って一般の乾電池の電圧はかなりいい加減なので，新品のときを考慮して20Vから12Vくらいまで電圧が変化しても安定して動くことを目指しています。

あれやこれやといじりながら，カット&トライを何度かやってこれでよしとなったのが，121ページにある図1です。信号は原則として左から右へと流れていきます。

まず最初がこの乱数発生器の命ともいえる雑音源です。トランジスタのコレクタとベースを直結してGND(0V)に落としエミッタは470kの抵抗で電源に接続され，ほんの少しだけ逆方向に電流が流れるようにしておきます。470kとやや大きめの抵抗にしているのは電流が増えるにつれて雑音レベルが下がっていく現象が見られたからです。

さすがに電源電圧が2倍近く変化すると，流れる電流も雑音の発生ぐあいも変化してきます。ノイズ発生用のトランジスタに流す電流が増えていくと，次第にノイズ出力が減っていって，最後にはごく小さな波形になってしまいました。雑音源として使うには，あまり電流を流さないようにしなくてはならないようであることがわかったので，大きめの抵抗にして20V以上でも安定して雑音が発生されることを確認しておきました。

信号を増幅する

ここで発生された雑音を次段のトランジスタで増幅します。間にある0.1μFのコンデンサは，10V前後の直流成分をカットし，交流成分である雑音信号だけを次段のトランジスタのベースに与えるためのものです。バイアスはいちばん簡単な固定バイアスで，コレクタ電流が1mAくらいになるように計算します。このとき，トランジスタのh_{FE}(直流電流増幅率)が問題となります。私が大量に買い込んだトランジスタは2SC693Eと最後にEがついたものだったのですが，一応，電源が12Vでh_{FE}が80とコレクタ電流がいちばん少なくなる条件で1mAくらい流れるように計算します。h_{FE}を80とし，コレクタ電流が1mAならベース電流は1/80mAです。ベースの電圧はほぼ0.6Vと固定ですから残りの12−0.6=11.4Vが抵抗にかかる電圧です。あとはオームの法則で，

$$R=E/I=11.4/(1/80)=912(k\Omega)$$

ここから，抵抗の値は1M(=1,000k)Ωくらいでよかろうということになりました。ちなみに，コレクタ電流が最大になるのは電圧が20V（一応20Vを上限と考えておきます)，h_{FE}が最大（160）のときです。このときのベース電流は，

$$I=E/R=(20-0.6)/1000=0.0194(mA)$$

コレクタ電流はこのh_{FE}倍ですから0.0194mA×160=3.10mAとなります。

これだけ動くと，今度はコレクタの負荷抵抗をどのくらいにするか気になります。教科書的にいくならI_C-V_{CE}特性図に負荷線を引いて考えるのでしょうが，そこまで真剣に考えるほどの増幅回路ではないので，コレクタ電圧が5～6V以上ならよいという方針で簡略設計で間に合わせます。電源電圧が低く12V，h_{FE}が160のときにコレクタ電流は，

$$(20-0.6)/1000\times160=1.8(mA)$$

くらいです。ここで，コレクタ電圧が6Vとすると抵抗値は，

$$R=E/I=6/1.8=3.3(k\Omega)$$

となります。

h_{FE}が小さく，コレクタ電流が1mAのとき，この抵抗の両端の電圧は3.3Vと，少少小さく，入力の雑音が大きいときには出力波形の頭がクリップする可能性もありますが，別にオーディオアンプにしようというのではありませんから構わないでしょう。

波形の整形

この増幅段で大きくなった雑音はやはりコンデンサを通って次段のオペアンプに入力され，デジタルっぽい波形に整形されます。オペアンプをこのように使うと，コンパレータ（電圧比較器）として動作します。+入力と−入力の電圧が比較され，+側のほうの電圧が高ければ出力は+電源電圧くらい(仮に1としておきます)，−入力のほうが高ければ−電源電圧付近の電圧（仮に0としておきましょう）になります。つまり，どちらの電圧が高いかということによって2値信号（デジタル信号）に変換できることになります。今回はどちらの入力も220kの抵抗で，電源を1対1に分割し，片方にいましがたトランジスタで増幅した雑音を放り込みます。通常，入力は抵抗の誤差くらいの電圧差しかありませんから出力はぎりぎりのところで，どちらかに転んでいる状態です。ここで，片方が雑音によって振り回されるために，出力が1になったり0になったりするわけです。

さて，ここでもあれこれ考える必要がありました。まずオペアンプの選択ですが，これは秋葉原での買い出しのときに見つけたLM318です。オリジナルはナショナルセミコンダクタですが，フェアチャイルド，テキサス，日電，AMDなどもセカンドソースを作っているので入手は容易であると思います。LM318は帯域幅が15MHz，スルーレート（出力の変化する速度）が50V/μsとなかなか高帯域で高速なオペアンプで，今回のように周波数が高めのデジタル信号(方形波）を得ようというのには好都合です。

入力信号があまり高い周波数になったらうまくないような感じなので，ローパスフィルタでも入れたほうがよいかなと一瞬思ったのですが，よく考えてみるとそれほど高い周波数の成分は入ってきそうにありません。前段の増幅に使った2SC693Eのコレクタ電流が1mAくらいのときのfT(遮断周波数，利得帯域幅積）は100MHzということです。トランジスタの増幅率は周波数が高くなっていくと，あるところで下がり始め，それ以降はだいたい6dB/oct（周波数が2倍になれば増幅率が半分）の割合で落ちていきます。そして増幅率が1，すなわち増幅器として使えなくなる周波数がfTです。低周波での増幅率が100程度と考えると，だいたい1MHzあたりで増幅率がずるずると落ち始めているというわけで，いわば1MHzのローパスフィルタを通したようになっているのです。ですから，オペアンプにはそれほど高い周波数の信号は入らないと考えて構わないであろうと考えられます。

電源もちょっと気になったところです。オペアンプは通常，2電源で使うのですが今回は片電源で使っています。2電源とみなすには，たとえば2個の乾電池のつなぎ目をGNDとすればよいのですが，そのGNDをオペアンプのどこにつなぐのかといわれれば，どこにもつなぐところはありません。結局片電源となにも変わらないのでそのまま使っています（入力を1対1に分割したのは2電源の場合のGNDの電位のイメージであるというのが真相です）。LM318の動作電圧は±5Vから±20Vです。片電源に換算すれば10Vから40Vということです。今回，電源は+12V以上と考えていますので十分でしょう。

信号レベルの変換

さて，このオペアンプの出力は方形波信号であり，デジタル信号風であるとはいえ電圧レベルは電源電圧近くまでとかなり高いので，このまま5Vという低い電圧の世界で動いているデジタルICに接続するわけにはいきません。また，一般にオペアンプの出力は電源電圧よりも2Vから3V程度の

電位差があるのがふつうです。オフセットを3Vと考えれば0のときには3V，1のときは電源電圧が12Vなら9Vの出力となります。

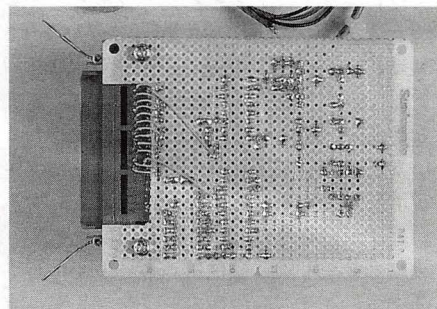
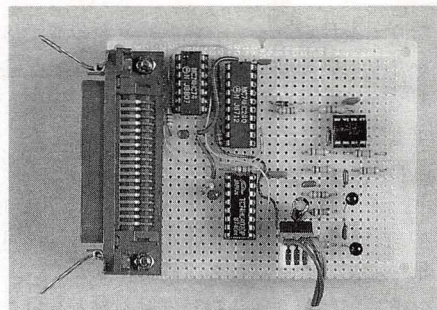
コンピュータとつなぐにはこれをデジタルICの論理電圧レベルに変換しなくてはならないのです。デジタルICのうち，TTLのH（High）レベルは2V以上，Lレベルは0.8V以下（0.8Vから2Vの間の電圧はH，Lのいずれと受け取られても文句はいえない）で，C-MOSの場合にはHが電源電圧の約2/3（電源が5Vなら3.5V）以上，Lは電源電圧の約1/3（同じく1.5V）以下で，いずれも電源電圧を越えないこと（Highのときに0V未満にならない）が条件です。

Hレベルのほうは，電源電圧が2倍近くも変わることから電源電圧が12VのときになんとかHレベルになるようにしておいて，高くなりすぎる分はツェナーダイオードを使ってリミットをかけることで処理することに決めました。問題はLレベルです。出力が3VというのはC-MOSのICを使うにしても高すぎます。

出力をレベル変換するのにトランジスタをスイッチ代わりに使うというのは定石のひとつですが，今回のような場合には少々不向きです。というのは，トランジスタをスイッチ代わり，すなわち飽和領域で使った場合，そのスイッチング速度というのはあまり速くないのです。高速スイッチング用のものでも，デジタルICに比べると見劣りがする場合がほとんどです。このため，綺麗な方形波を入れても出力は角が丸まった，なんとも情けない波形になり，細かいパルスは波打つだけになってしまいます。

かなり高いスルーレートを持つLM318の出力波形はせいぜい1MHz程度の入力信号に対しては相当綺麗な方形波にしてくれることが期待できるのに，レベル変換のために波形がなまるのは面白くありません。そこでLレベルを綺麗に0V近くまで落とすことはあきらめて，出力を適当に抵抗で分圧することで，デジタルICがLレベルと認識できるくらいの電圧にしてしまうことにしました。Lレベルのことだけを考えるならなるべく電圧を小さくする方向に分圧したいところですが，それとともにHレベルのほうも落ちてくるので注意しないと信号が全部Lレベルになってしまいます。

微調整を加える

出力には4.7Vのツェナーとコンデンサが入っています。まず，ツェナーの消費する電力を計算しておきましょう。出力が20V程度まで上がってしまったとすると，4.7kの抵抗に流れる電流は（20－4.7）/4.7＝3.26（mA）です。このうち，3.3kの抵抗に流れる分は4.7/3.3＝1.42（mA）ですから残りの3.26－1.42＝1.84mAがツェナーに流れる電流です。したがってツェナーの消費電力は1.84×4.7＝8.65（mW）となります。今どきのツェナーは小さいものでも200mWくらいですから余裕しゃくしゃくといったところです。この余裕を見込んでつけたのが4.7Kに並列につないである47pFのコンデンサです。

抵抗で分割して取り出した波形は，教科書どおりの世界であれば入力信号と完全に対称になるはずなのですが，方形波が相手の世界では現実はそれほど簡単にはいきません。部品の足や基板がある以上，現実にはあちこちに細かい容量が存在します。おまけに接続されている74HCシリーズは入力は電流こそほとんど流れないとはいうものの，10pFくらいの入力容量があります。つ

乱数源を求めて

物理乱数を使うとはいっても，まさかコンピュータにコインを投げさせて1/0を決めるような方法は使えません。電気で動くコンピュータが相手ですからやはりランダムな電気信号（ノイズ）を用いることになります。手近なノイズ源としては，デバイス（半導体だけでなく，抵抗なども立派なノイズ源です）の発生するノイズ，FMラジオの局間ノイズ，変わったところでは放射線のカウントを行う方法などがありますが，発生量や安定性，信号レベルの兼ね合いなどから，使いものになる信号は限られてきます。

実際に作られているものを見ると，たとえばHITACではノイズ発生源としてツェナーダイオードの発生するノイズを使ったようです。ATARIではトランジスタのノイズを使っているという話を聞いたこともあります。後者についてはよくわかりませんが，前者で使われたツェナーダイオードについては電子デバイスの中でも割と大きなノイズを発生するというのはアナログ回路屋さんの世界ではよくいわれることです。このノイズは通常は単なる邪魔ものにすぎませんから，対策しておけとは書いてあっても，それなら対策しないとどのくらいの電圧のノイズが出るのかということを述べたものは見当たりませんでした。もっとも，ノイズを積極的に使おうなどとはふつうは考えませんからしかたありません。

HITACで使われたツェナーダイオードを乱数源にする方法は雑音を2値信号に変換するだけですから（ツェナーダイオードはせいぜい20円から30円くらいということからも），かなり手軽にできそうな気がします。雑音レベルのデータがないのが気がかりではありましたが，試しに作ってみることにしました。

できた回路はツェナーダイオードに軽く電流を流しておいて，トランジスタ1個で増幅してからリミットをかけるだけのごく単純なものです。ところが，電源を入れてもうんともすんともいいません。おかしいなあと思いつつ，少しずつ電圧を変えていくと，電源がある電圧のときにほんのわずかな出力が見られるといった程度であることがわかりました。おおもとのツェナーダイオードのところで見てみると，信号レベルが低すぎるせいかほとんど雑音らしきものが見られません。かなりいい加減な測定ですが，電圧レベルはせいぜい大きくても100μV以下という感じです。

確かに，抵抗などのノイズと比べれば桁違いに大きいことは間違いありませんが，こんなに小さい信号が相手では外乱に対してあまりにも弱いですし，実装や配線のノウハウが必要になるため，個人的には面白くても製作記事としては少々問題があります。手持ちのツェナーダイオードをとっかえひっかえ試してみたのですが，たいして変わりませんでした。ツェナーダイオードが雑音源といわれてきたことから，低雑音を意識して作られてきているのかもしれませんが，いずれにしても，乱数源を変えなければならないことは確かなようです。

ツェナーダイオードが雑音源となるのは，その特性，すなわちある電圧（ツェナー電圧）に達すると突然電流が流れ始め，それ以降は電圧をほんの少し変えるだけで，流れる電流が大きく変化します。見方を変えれば電流が大きく変化しても，ツェナーダイオードの両端の電圧はほとんど変化しないということになります。これとよく似た特性を持った現象の見られるものをみつければ雑音源として通

図1　回路図

図2　端子表

用するのではないかといろいろ考えてみました。

NPN型のトランジスタでは，ベースからエミッタに流す電流をほんの少し操作するだけで，コレクタからエミッタに向かって流れる電流が大きく変化します（PNP型なら電流の流れる向きが逆になります）。このとき，エミッタからベースに，すなわち通常と逆の方向には電流は流れないとふつうは教わるわけです。実際にテスタなどで抵抗値を測っても逆方向の抵抗値はきわめて高く，確かにほとんど電流が流れないようです。

それではちょっと意地悪をして，逆電圧を少しずつ上げていきながら，トランジスタのエミッタとベースの間に流れる電流を測定してみます。すると，ある電圧に達するとそれまでほとんど動かなかった電流計の針が急激に振れるようになります。電圧を落とすと，また電流が流れなくなります。このような現象を降伏と呼んでいます。通常，よく使われる小信号用のトランジスタでは10V程度の電圧で降伏を起こします。テスタで抵抗を測るときは内蔵の乾電池の電圧くらいしかかけないので，降伏電圧には達しておらず，したがって電流もほとんど流れなかったというわけです。

ここをもう少し調べてみると，NPN型のトランジスタではコレクタをオープンにして測ると，この部分で電圧が上がるほど電流値が減るという負性抵抗を示し，コレクタをベースと直結すると電圧を上げるに従って急激に電流値が上昇していきます。これはまさしくツェナーダイオードそのものです。ただし，トランジスタをこのように使うことはいわば裏技ですから，降伏電圧は同じ型番のものを同じ条件で測定してもかなりばらつきがあります。当然，こんな異常な使われ方をした場合に発生するノイズ対策のことなど考えられているわけはありません。きっと大きなノイズが出てくれると信じて基板の改造を始めました。

トランジスタは200個500円という値段に引かれて衝動買いしてしまった2SC693がありますので，まあ失敗してもともとくらいの感じで実験してみました。ベースとコレクタを直結して2本足にして，ツェナーダイオードと交換します。電源電圧も最低10Vはないとトランジスタが降伏を起こさないので，増幅段の定数も計算しなおしです。

どうにか改造が終わって，再起動です。出力を監視しながら電源電圧を徐々に上げていきます。6V，7V，まだまだなにも出てきません。おっかなびっくり電圧を上げて，9V付近を越えたとたん，いきなり針が振り切れてしまいました。「やばい！　壊したかな」と，あわてて電源を切ったのですが，別になにも起こった様子もありません。今度は電流を監視しながら電圧を上げていってみます。やはり9V前後で電流が流れ始めます。しかし，異常と思えるような流れ方ではありません。

もう一度出力を調べてみると，1V以上もあります。「ありゃ？　発振してしまったのかな」と，雑音源のトランジスタの両端にコンデンサを入れると，すっとレベルが落ちていきます。離すとまた1V以上の出力です。どうやらツェナーダイオードのときとは比べものにならないほど大きな雑音が出ているようです。それにしても単位がVとは思いませんでした。せいぜい100mVくらいと考えていたのですから，振り切れるわけです。とりあえず，これでトランジスタからよいノイズが取り出せそうです。

まり，ツェナーにさらに並列にコンデンサをつないだような状態になっているのです。オペアンプの出力が0になるときは，このたまった電荷を抵抗を通して抜かなくてはなりません。逆に1になるときは抵抗を通してチャージしてやらなくてはならないわけです。このため，方形波の角がなまり，立ち上がりや立ち下がりがなまってきます。これを補償する目的で入れたのが47pFのコンデンサです。立ち上がりや立ち下がりのときには抵抗だけでなく，このコンデンサを通して一気に電荷のチャージ/引き抜きをやってやろうというものです。

このコンデンサの容量が少ないと波形のなまりが残りますし，容量が大きすぎると立ち上がりや立ち下がりがいったん行きすぎたあと今度は跳ね返るようにしばらく暴れる現象(オーバーシュート/アンダーシュート）が出てきます。この暴れが大きく，立ち下がりのあとの跳ね返りをデジタルICにHレベルへの変化と認識されてしまったり，立ち上がりのあとをLレベルになったと判断されては面白くないので，このコンデンサの容量は注意して選ぶ必要があります。

計算式もあるのですが，結局は波形を観測しながらカット&トライで決めるほうがよいようです。私は手持ちの都合で27pFにしたのですが，少し容量が足りなかったようです。ついでにやった実験では100pFでは少々補償のやりすぎのようでしたので，回路図では47pFとしておきました。もし，シンクロ（ストレージスコープがあればベストですが）が使えるようでしたら，波形を見ながら調整するとよいでしょう。

確率の調整

さて，このようにしてトランジスタで発生した雑音がようやくデジタルICのレベルにまで変換されました。ただし，このまま読み取ると，1と0の発生する確率がノイズの波形に依存してしまうことになるためフリップフロップで分周して1と0の発生確率が1対1になるようにします。フリップフロップを使うということから，クロック入力に先ほど分圧したオペアンプの出力がつながれることになるのですが，作り方によってこの波形がなまったり，多少補償をやりすぎてオーバーシュートなどが出てしまったとしてもそれを誤って拾うことがないようにするため，シュミット特性を持っているHC240を通しています。

このフリップフロップの出力は当然のことながら常時変化しています。これをパソコンに読み取らせる場合，そのままにしておくと読んでいる途中でデータが変化する可能性が非常に高くなるので，安定した読み取りを保証するために，この出力をラッチする回路をつけておきます。つまり，パソコン側から「ハイ！」と掛け声をかけた瞬間のデータをラッチさせておいて，あとからゆっくりと読み出しにいこうというわけです。ラッチしたデータは次の掛け声があるまで変化しませんから気楽に読み出せます。出力はバッファである，HC240を通して出しておきます。

「掛け声」に相当する信号もHC240を通します。入力に直列に入っている抵抗は入力保護です。C-MOSデジタルICの場合，入力電圧が電源電圧よりも高くなると入力ピンからICの電源に向かって大きな電流が流れることになります。パソコンの5Vと乱数発生器の5Vが共通でない以上，なんらかの電位差はどうしても出てきます。74HCの場合には入力に保護回路が入ってはいるのですが，ICの保護回路は万が一のためにあるのですから，これに期待するというのも危ない感じです。とはいってもツェナーを使うというほどのものでもなさそうなので，ここは抵抗を入れて，過電流を防止しています。抵抗で5Vとつないでいるのは，入力がオープンになったときに入力がふらふらして，ラッチアップを起こさないようにすることと，パソコン側の出力ICがTTLであったときにはHレベルがやや低くなりがちなので，Hレベルを保証するために入れてあります。

74HC4020と，LEDは動作には直接関係のない，おまけ回路です。基板に少々隙間が残っていたので，乱数が発生されているかということと，パソコンの読み込みが行われているかを目で確認できるように入れてみました。

表1　部分一覧

	型　番	数量	単　価
トランジスタ	2SC693E(またはF)	2	30
発光ダイオード	TLR102など	2	30
3端子レギュレータ	78L05	1	80
オペアンプ	LM318	1	320
デジタルIC	74HC240	1	150
	74HC4020	1	220
	74HC74	1	100
ツェナーダイオード	RD-4.7A	1	15
電解コンデンサ	10μF/50V	1	25
セラミックコンデンサ	0.1μF	8	15
	47pF	1	15
抵　抗	1k	1	5
	3.3k	4	5
	4.7k	1	5
	10k	1	5
	220k	4	5
	470k	1	5
	1M	1	5
コネクタ	アンフェノール36ピン	1	800
	電源供給用	1組	40
電池スナップ	006P用	2	20
乾電池	006P	2	100
プリント基板	ICB-504EG(サンハヤト)	1	300
ICソケット	14ピン	1	15
	16ピン	1	15
	20ピン	1	20
計			2760円

そして接続

さて，あとはこれをパソコンのどこから読ませるかということだけです。ビット数が多ければパソコン本体の拡張スロットに差さるように設計するのですが，今回は1ビットの乱数ですから，入力端子が1ビット，さらに「掛け声」のための出力端子が1ビットあれば十分ですので，外づけにすることにしました。候補はジョイスティックポートとプリンタポートですが，ほとんどの機種についていて，しかもコネクタが(プリンタ側で）統一されているのでプリンタポートを使うことにします。プリンタ側はセントロニクスに準拠しているものがほとんどですが，パソコン本体のほうは必要な信号以外は無視してしまっていたりするので，必ず使われている信号ということからBUSYを乱数データ入力，STBをラッチ信号（掛け声）に使うことにしました。

組み立て

精密さを要求される回路ではないので，特に注意しなくてはならないようなところはありませんが，ちょっとしたコツのような部分について触れておきましょう。

基板はカッターでパターンを切りながら作っていきますが，もし彫刻刀のような新兵器を持っているならV型のものを使うとよいでしょう。なお，勢い余って手を切ったりしないように注意してください。

基板を2つに割るときには，表と裏から軽くキズをつけておいてから机のへりなどで曲げれば簡単に割れます。もともと，2.54mmピッチで穴が開いているという状態は，基板にミシン目が入っているようなものですから，無理に力をかければ割れやすくなっているのです。カッターでキズをつけるというのは，この割れやすい状態に癖をつけてやるという作業なのですから，それほど深く切り込む必要はありません。

カッターは一般の紙工作用のもので十分です。私は以前パズルを作ったときに買ったものがあったのでそれを使いましたが，カットのときには結構力をかけるので，これから買うのでしたら刃をねじで固定する方式のものがよいでしょう。

カッターでパターンを切り離すときにはパターンに傷をつけるだけでなく，適当な幅で剥がすようにします。切り口がV字になるようなイメージで，両側から切るとペロッとパターンが取れます。

部品配置などが決めづらい人は写真を参考にしてください。アナログ部分は，回路図とよく似た配置になっているでしょう。こうしておくとのちのち，波形を見たり調整するときにどこを観測しているのかわかりやすくて楽です。

デジタルICの電源ピンとグラウンドの間には0.1μFのコンデンサを入れておきます。これは電源ラインのノイズを抑えるためのもので，回路図には出てこないのです。工業用の製品などではICの電源ピンごとに入れている例もあります。われわれアマチュアが作り，比較的よい環境で使われるものの場合にはそこまでする必要はありません。

テスターを持っておられるようでしたら，ソケットにはICを差し込まないで，消費電流を測っておきます。もし100mA前後も流れたらどこかおかしいので，電源を抜いてチェックしなおしてください。次にICの電源電圧を見ます。デジタルIC（74HCシリーズ）のGNDとVcc端子の間の電圧は5Vになっているはずです。もし，大幅に違うようでしたら，3端子レギュレータまわりなどを見直してください。

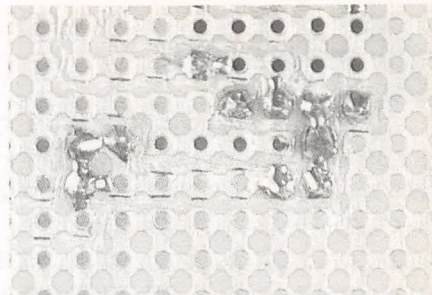
OKとなったら，いったん電源を切ってからICを差し込みます。足は広がり気味なので机の上などで真っ直ぐに整形しておきます。足先ではなく，足の根元近くを折り曲げる気持ちでやると綺麗にできます。

テスターがない場合はじっくりと基板をチェックして，ショートなどがないかチェックして，勇気をもって一気に電源ONとなります。電源を入れると，74HC4020につけたLEDがせわしく点滅を繰り返しているはずです。

調整

あちこち気をつけたつもりなので，調整する場所はほとんどありませんが，テスターを持っていたら，増幅段のトランジスタのコレクタ電流を測っておきましょう。測定は，パターンを切ったりしなくても，コレクタにつないだ3.3kの抵抗の両端の電位差を測って行います。だいたい1～2mA程度流しておく予定ですから3～7V程度になるはずです。あまりにもはずれているようでしたら，ベースについている1Mの抵抗を調整してください。もし，シンクロスコープなどが手近にあるようでしたら，各部の波形を観測しておくとよいでしょう。ストレージスコープなどがあればオペアンプの次段のHC240の入力波形から47pFの調整が容易に行えるでしょう。

さて，物理乱数である以上，発生頻度が問題ですので測定してみました。シンクロ，できたら周波数カウンタで，HC4020の3番ピン（LEDをつけた端子）の周波数を測り，それを2の14乗倍（16384倍）すればおよその値は知ることができます。

平均の発生頻度よりも極端に短い周期で取り込むと同じデータが続く場合が多くなりすぎて，とても乱数とは呼べなくなってしまいます。私も試作段階を含めて，何度か測定してみましたが，だいたい乱数出力（HC74の9番ピン）の周波数は平均すると500kHzから1MHzの間であるようです。ランダムな値を得るにはこれよりも十分に長い周期でサンプリングしなくてはなりません。できたら，100倍，すなわち100～200μsくらい間をおきながらサンプリングしたいところですが，1ビットを得るたびに200μs，すなわち0.2msかかるというのは，用途によっては遅すぎるかもしれません。そのときは実験しながら，読み込むインターバルを調整してください。

配線を切り出す

Z80ではIN/OUT命令に11サイクル（Cレジスタ間接なら12サイクル）かかります。STBを上げ下げして，データを読み出すだけで，OUTが2回，INが1回の計3回で33サイクルかかります。一般的な4MHzのZ80ならこれで1/4 ×33＝8.25(μs)です。68000の場合はデータ移動に各8サイクルで仮に10MHzなら結局2.4μsとなります。今回のサンプルのようにBASICから使うときはまったく意識する必要はありませんが，アセンブラで書いたプログラムから使うときはちょっと頭の隅にでも置いておいてください。

部品集め

回路はできましたので，あとは部品集めと組み立てです。部品とその値段（概略）は，部品表に載せたとおりですが，コネクタは少々みつけにくいかもしれません。私はジャンク屋で1個100円でみつけたものを使ったため，型番などがわかりません。店で探してもらうときは「基板取りつけ用のアンフェノールの36ピンのメス」といえば（およそ日本語とは思えませんが）わかってもらえるでしょう。

また，トランジスタの2SC693はEタイプは製造中止のような感じがするので，部品表ではFも指定しています。もし，GやHしかない場合には，コレクタ電流が大きくなりすぎるので，ベースにつないでいる1Mの抵抗をもっと大きいものと取り換えて，コレクタ電流が数mAになるようにしてください。693がない場合にはほかの2SCタイプの高周波の小信号用トランジスタならだいたい使えると思いますが，実験したわけではないのでなんともいいきれません。

基板は専用プリント基板を作ろうかとも思ったのですが，エッチングはあまりやりたくないので，なにかよさそうなものはないかと，パーツショップをうろうろとしていたらパターンがメッシュ状に連なっているICB-504EG（サンハヤト製）がみつかりました。この基板の銅箔は買った時点では全体がつながっています。これをそのままグラウンドラインにして，信号ラインのところだけをまわりから切り離すようにして使えば，電気的には専用プリント基板並みのものがカッター一丁でできそうです。しかも値段が300円と，一般的なひとつ目のものと大差ありません。

また，5Vの3端子レギュレータは，私の基板ではこれまたジャンク屋で4個100円で入手した7805（1A）を使っていますが，電流はほとんど食わないので，新たに買う場合は小型で安価な78L05で十分です。

LED（発光ダイオード）は小型のものならなにを使ってもよいでしょう。昔はLEDといえば赤と緑しかありませんでしたが，今では黄色やオレンジ，青などもあって，なかなかにぎやかです。趣味にあわせて選んでください。

読み取りソフト

完成したら，パソコン本体と接続します。STB信号を０にして１に戻すという操作を繰り返し行うと，その都度74HC74につないだLEDがランダムに点滅するはずです。うまくいっているようでしたら次に，STBにパルスを送ったあと，BUSY信号を読み出してみます。１になったり０になったりするのが読み出せれば成功です。

ついでですからMZ-2500とX1用に，BASICで簡単なプログラムを組んでみました。１回読み出すたびに画面に点を打っていくものと，最初に49カラム目から始め，読み出したデータによって左右に♯マークを動かし，端までいったら再び49カラム目からスタートするというもの，そして円周率を求めるものの３つです。♯マークが動くものは酔っぱらいの千鳥足を連想させることから酔歩と呼ばれるものです。実行して遊んでみてください。

さて，せっかく作った乱数発生器をこれで終わらせるのはちょっとしのびありません。近く，この乱数発生器を使って遊んでみることにしましょう。

だいぶ前にIC関係が一気に品不足になり，ふつうなら30円くらいのICを200円という値段で売りつけられて頭にきて以来，ふて寝を決め込んでいたハードの虫がむずむずと動き出してしまいました。

何年ぶりかで秋葉原のパーツショップやジャンク屋を真剣にまわってみると，10年も前とあいも変わらぬその値段に思わず引かれて気がついたときには工具やら細々した部品やらをあわせて１万円以上も買い込んでいました。最近はプログラムばかりやっていたけれど……。けどね，それでもね，私はハードが好き！

リスト１　サンプルプログラム(MZ-2500/X1)

```
1000 '=================================
1010 '= 乱数発生機チェック用プログラム =
1020 '=                               =
1030 '=================================
1040 INIT "crt2:640,200,16"     :' INIT:WIDTH 80<---X1
1050 CLS 2:CLS                  :'
1060 INPUT "0：乱数パターン表示 1：酔歩 2：円周率 ";A
1070 IF A=0 THEN 1330
1080 IF A=1 THEN 1140
1090 IF A=2 THEN 1420
1100 BEEP:GOTO 1060
1110 '
1120 ' 酔歩：どちらにいく確率も同じはずです
1130 '
1140 L=0:R=0
1150 CLS
1160 FOR I=1 TO 100
1170    X=39
1180    GOSUB 1580
1190    LOCATE X,0:PRINT
1200    IF A=0 THEN X=X-1 ELSE X=X+1
1210    IF X=0 THEN L=L+1:LOCATE 0,10:PRINT "左 = ";L;"回";:GOTO 1250
1220    IF X=78 THEN R=R+1:LOCATE 0,11:PRINT "右 = ";R;"回";:GOTO 1250
1230    LOCATE X,0:PRINT "#";
1240    GOTO 1180
1250 NEXT
1260 LOCATE 0,15:PRINT "END"
1270 BEEP
1280 END
1290 '
1300 ' 乱数のグラフィック表示：
1310 '     明らかにわかるような周期がでていないことを確認してください
1320 '
1330 CLS:CLS 2
1340 FOR Y=0 TO 199
1350    FOR X=0 TO 639
1360       GOSUB 1580
1370       IF A=1 THEN PSET (X,Y),7
1380   NEXT
1390 NEXT
1400 BEEP
1410 END
1420 CLS
1430 '
1440 ' 乱数によって的撃ちを行い，１／４円のなかに入る確率から円周率を求めます
1450 '
1460 PI=0
1470 FOR I=1 TO 10000
1480    GOSUB 1670:X=B
1490    GOSUB 1670:Y=B
1500    IF X*X+Y*Y<=65025 THEN PI=PI+1
1510    LOCATE 0,0:PRINT USING "#.####";PI/I*4
1520 NEXT
1530 BEEP
1540 END
1550 '
1560 ' 乱数発生機からのデータを読みだし，変数Ａに入れて返します
1570 '
1580 'A=rnd(1):if A=.5 then 1580
1590 'if A<.5 then A=0 else A=1
1600 'return
1610 OUT &HFE,&H80:OUT &HFE,0       'OUT &H1A02,&HF:OUT &H1A02,&HE
1620 A=INP(&HFE) AND 1              'A=(INP(&H1A01)/8) AND 1
1630 RETURN                               以上Ｘ１用でした
1640 '
1650 ' ８ビットの乱数を作成します． 値は変数Ｂに入れて返します
1660 '
1670 B=0
1680 FOR ZI=1 TO 8
1690    GOSUB 1580
1700    B=B*2+A
1710 NEXT
1720 RETURN
```

●ハードウェア工作編

X1turbo用バンクメモリボードの拡張

512Kバイトの誘惑

Kamon Masato
華門 真人

NEW Z-BASIC に付属のバンクメモリボードを256Kバイトに拡張してみましょう。といっても，メモリを買ってハンダづけするだけと作業はとっても簡単です。あなたもお手持ちのX1turboを手軽にシステムアップしてみませんか。

X1turboの64Kメモリボードをご存じだろうか。そう，Z-BASICでMMLなどに使われている例のメモリである。X1turbo の発売当時からあったX1turboには将来的に512Kバイトのバンクメモリがサポートされているというアナウンスがようやく実現したわけである。増設バンクメモリは，X1turboZII以降でバンク0,1が標準装備になっている。もちろん初代turboなどでもNEW Z-BASIC(CZ-141SF)を買うことによってもれなくこのバンクメモリボード(バンク0,1)がおまけ(?)でついてくるからまったく問題ない。

ここで問題なのはNEW Z-BASIC を買ったときについてくるバンクメモリボードである。このボードは0000H～7FFFH のバンクメモリがバンク0，バンク1の2つあるはずであるから，結局全体の容量は32Kバイト×2の64Kバイトのはずである。ところがこのボード，実はほんのちょっとの改造で容量が256Kバイトになるという噂がある。これが本当であれば実に美味しい話である。果たして噂は本当なのであろうか。

美味しい話

すでにNEW Z-BASICを手にした諸兄はご存じと思うが，CZ-141SF(NEW Z-BASIC) 用のバンクメモリボードには配線はしてあるけど石が差さっていない個所が6カ所ほどある。これはどう見てもメモリ増設用にしか見えないのである。実際，ボードには，

TMS4464-12
orMB81464-12
orLH2464-12

という石の名前が書いてある。464といった数字の並びからわかるように，これは4ビット×64K=256Kビット(=32Kバイト)DRAMの型番である。空きエリアが6個あるわけだから全体の容量は64Kバイト（元からある分）+32Kバイト×6=256 Kバイトということになる。う～ん，なんとも美味しい話ではないか。

どう見てもすべて配線済みだし，なんとご親切にもパスコンまで差さっている。これでは動かないほうがおかしいといえよう。メモリICをここに差すのは間違いないとして問題はジャンパやパターンカットなど，ほかの配線を操作しなければならないか否かということであろう。結論からいえば，手を加える必要はまったくない。RAMをハンダづけするだけでちゃんと動く。

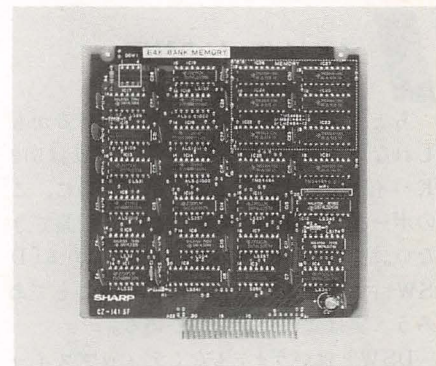
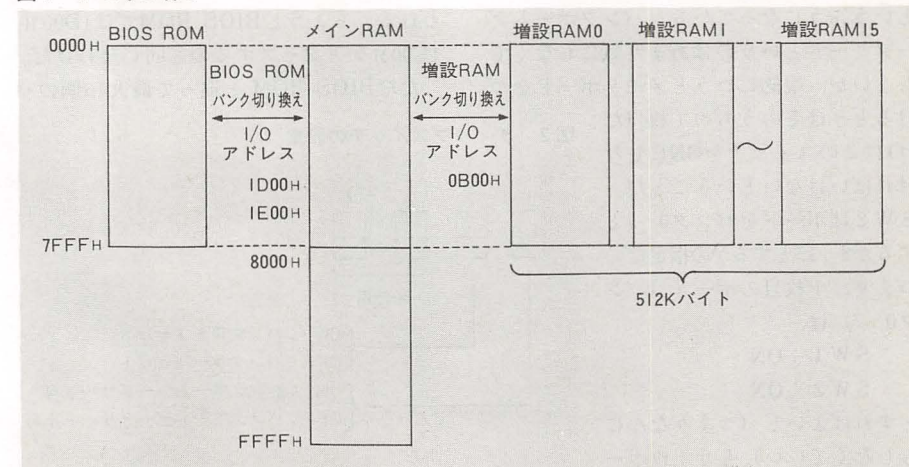
バンクメモリ3分間クッキング

材料
- 256KビットDRAM　　6個
  TMS4464-12
  or MB81464-12
  or LH2464-12

お店に行ってこれを6個といえばふつう通じるはず(1個1200円程度)。
- バンクメモリボード　　1個

CZ-141SFに付属のもの。これがなければお話にならない。
- ソケット　　6個

上記DRAM用のもの。よほど腕に自信のある人以外はソケットを使ったほうが賢明だろう。石を熱破壊する心配がないし，必要とあらば石の使い回しもできる。
- ハンダゴテ，ハンダ少々など

1）バンクメモリボード（以下ボードと略す）にソケットを6個ハンダづけする（ボード上の白点線で囲まれポッカリと空いている部分だよ)。終わったらスロットに入れて無事起動することを確かめる。

2）大丈夫ならソケットにDRAMを注意深く差す。差すときは石の方向に注意。石は左右対称じゃないぞっ！　半円形の切り欠きをほかの石と同じ方向に合わせること。

3）動くように念じつつボードを本体のスロットに差す。いやぁ，おめでとう。これで256Kバイトバンクメモリボードの誕生だ。新しい生命(?)の誕生に乾杯！

とまぁ，本当に3分間写真並みの簡単さでできてしまうのだ。

メモリが6個揃わないときは手元にあるだけ順番につけると1バンクずつ拡張できる。ひとつつけては父のため，2つつけては母のため……。18ピンの石が6個だから

図1　メモリの様子

ハンダづけは108つ。除夜の鐘でも聞きながら，作業を進めるとよい。

ただ，ボードの動作に不安を覚える人がいるだろう。しかしその点は大丈夫。シャープ発その筋によると，ちゃんとした動作をするということであった。もちろん自分で改造した以上，責任はすべて自分にある。シャープは一切保証してくれないであろうからそのつもりで（ハード改造にリスクはつきものなのだよ）。

メモリボードの設定

ちょっと疑問を感じている人がいるかもしれない。X1turboのバンクメモリは512Kバイトまでサポートされているのに，このボードでは256Kバイトまでしか拡張できない。注意深い人ならボード左上にある「DSW1」と書かれた部分に気づいたことであろう。実はこれが鍵なのである。

DSWとはいうまでもなくディップスイッチのことである。そう，本来ならばここにディップスイッチが差さっているはずなのである。そしてそのディップスイッチでバンクを指定，というふうになるはずなのだが……，どうやら経費節減（？）のためかディップスイッチは差さっておらず，直接配線してある。

よく見てみると，左から1，2，3，4というふうにスイッチ(SW)が並んでいる(図3)。さらにSW1とSW2は配線によってつながっている（すなわちONの状態）ことがわかる。これらのスイッチの意味は，

SW1
　ON　：バンクポートデータリード可
　OFF：バンクポートデータリード不可
SW2
　ON　：バンク0～7設定
　OFF：バンク8～15設定
SW3，4：未使用

というふうになっている。バンクポートデータリードというのはあまり気にしなくてもよいが，端的にいうとメモリボードをつけるときはそのうちの1枚（だけ）はこのスイッチをONにしなければいけないということだ。SW2はボードをバンク0～7とするか8～15とするかの指定だ。つまり，1枚目のボード（バンク0～7）は，

　SW1：ON
　SW2：ON

とすればよいし（つまりなんにもしなくていい），もう1枚ボードを買ってそれをバンク8～15とする場合には，

　SW1：OFF
　SW2：OFF

とすればよいだけのことである。これはボード上の配線を2本切っておけばよい。

ちょくちょくバンク指定を切り換えたいという人は配線をカットした上でディップスイッチを取りつけ，ディップスイッチで設定すればよいし，固定したままでいい人はONならばそのまま(すでに配線がつながれている)，OFFならば配線をカットしてやればよい。

ほかに注意事項としては，動作の安定化のために本体内のI/Oスロットに取りつけてやること。また，ハード的に512Kバイト(バンク16個)しか持てないのだからボードを3枚以上差すな，というのは常識だろう。あ，そうそう，すでにバンクを2つ（バンク0,1)内蔵しているZII以降ではどうなるかだが，ちゃ～んと動く。ただし，バンク0,1は本体内のメモリを，2～7はボード上のメモリを使うことになる（この場合もスイッチ設定はSW1，2ともにON)。またバンク8～15はバンク0～7を取りつけたうえで取りつけること。

バンクメモリ活用術

さて，ともかくこれで256Kバイトバンクメモリボードは完成した。しかしハードはあってもソフトはない。現在のところソフトでバンクメモリをサポートしているのはZ-BASICのみである。しかしそのZ-BASICといえども単に変数領域としてしか使えない。やはり本格的にバンクメモリを使いこなすためにはおさだまりのマシン語へと走ることになりそうである。

X1turboにのバンク切り換えはI/Oポートの0B00Hをアクセスすることによって行われる。ちょうどBIOS ROMでは1D00H/1E00Hをアクセスするのと同じなわけだ。

ただBIOS ROMと違って最大16個のバンクの中からひとつのバンク（さらにプラスすることのメインメモリ）を指定しなければならないわけだから，BIOS ROMに比べればちょっとだけ面倒になってしまう（といっても全然たいしたことはない)。

具体的にはI/Oポートの0B00Hに図3のようなデータを出力してやればよいわけである。たとえばバンク1をセレクトする場合には，

```
LD   BC,0B00H
LD   A,00000001B
OUT  (C),A
```

とすればよいわけだし（この結果メインメモリの0000H～7FFFHがバンク1に置き代わる)，逆にメインメモリに戻す場合には，

```
LD   BC,0B00H
LD   A,00010000B
OUT  (C),A
```

としてやればOKというわけ。

バンクメモリを使うにあたってはメインメモリにはない特殊な性格を考慮しなければならない。まず，バンク切り換えのプログラムは8000H以降に置かなければならない。というのもバンク切り換えが行われた瞬間に0000H～7FFFHのメモリは入れ替わってしまうのであるから，バンク切り換えのプログラムが7FFFH以前に置かれているとその瞬間に元のプログラムを見失ってしまうからである（もっともこれを逆手に取るテクニックもある)。

同様にしてスタックを7FFFH以前に置くのも禁じ手である。ちょっと考えてみるとわかるが，PUSHしたデータとPOPしたデータとが違ってきてしまうといったようなことも起こりうるのである。

また，使用するにあたって切り換えを要するので，あまり切り換えばかりしていると必然的に実行速度が落ちてしまうのである。こういった点をふまえてやると，バンクメモリの使用法として次のようなものが浮かび上がってくる。

1）ほかのスタンドアロンなプログラムを置いておく

図2　ディップスイッチの設定

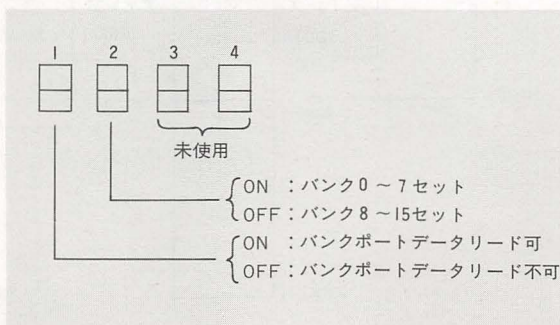
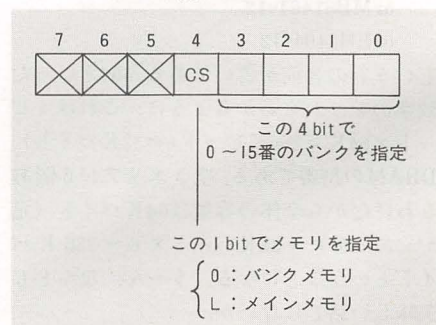



図3　バンク切り換え

2）サブルーチンの集まりとして使う
3）データエリアとして使う

1)は切り換えに時間がかかるならめったに切り換えなくて済むようにしてしまえということで，バンクメモリ内にまったく別のプログラムをおいてしまうということである。これを使えばメインメモリにturbo BASIC，バンクメモリ内にS-OSというきわめてエグイ取りあわせも実現できてしまう。この両者をうまく切り換えて使ってやればBASIC上で動くマシン語プログラムの開発なんかもラクラクである。すなわち，BASICからバンクを切り換え，S-OSを呼び出し，プログラム作成，アセンブル。プログラムができたら切り換えによってメインメモリ（BASIC）に戻り実行，不都合があったら再びS-OSにというきわめて能率のいいことができてしまうのだ。

これを実現するにはたいした手間はいらない。前に述べたバンク切り換えの手順によってバンク切り換えしてやればよいだけの話である。また，バンクメモリに入れるプログラムであるが，これはディスクから一度G-RAMにロードして，そこからバンクメモリに転送するといった方法が賢明だろう。JODAN-DOSをお持ちの方ならば，JODAN-DOSを利用するのも手である。

2)はバンクメモリにサブルーチンをまとめて入れてしまうという手である。発想的にはBIOS ROMとまったく同じ。ただ，バンク切り換えの時間を考えて高速を要求されるルーチンはメインメモリにといった使い分けをするといっそう効果的であろう。

使い方は基本的に（1）と同じである，プログラムやスタックの位置に注意してバンク切り換え，そしてサブルーチンコールといったぐあいである。うまい方法もある。リスト1がそれである。これはBCレジスタにコールしたい（バンクメモリ内の）ルーチンのアドレスを入れてRST28Hとするだけでコールできるという代物である（7FFFH以前からでもOK！）。ただし，002EHにコールしたいバンクのナンバー（0〜15）を入れておくこと。

これは先ほど7FFFH以前に切り換えプログラムを置くと，切り換えの瞬間にプログラムを見失うといったのを逆手に取った方法なのである。このプログラムのミソはバンクメモリ側にもプログラムをあらかじめ書き込んでおくことにあり，この方法はBASICがBIOS ROMを呼び出すときにも，使われている（RST18H）。なかなか面白い動きをしているのでパズルのつもりで解析してみると面白いだろう。

3)はいうまでもなくいちばん多い使われ方である。Z-BASICのVDIMはこれにあたる。これについては今さらつけ加えることもないだろう。単に容量が増えただけのことである。

最後に

さて，こうしてバンクメモリ増設およびその活用法について見てきたわけであるが，なぜもともと256Kバイトメモリとして売らなかったのかという疑問が残る。これはやはりおさだまりの経費節減というやつであろう。ま，とりあえず64Kバイトあれば十分，と考えたのだと思う。これで足りない人は自分で増設してください，ということだろう。その証拠があの異常なまでに丁寧な書き込みである。私はあそこまで親切に書き込まれたボードは見たことがない。

しかし残念なことに肝心の256Kビット DRAMがいまだに品薄でなかなか手に入らない（ここらへんにも原因があるかもしれない）。手に入らないとなると初めからつけておいてくれればよいのにと思ってしまうのも人情であるが，ともかくこのバンクメモリ，今まででは考えられなかった広い空間を手にできるのだ。DRAMが手に入ったらぜひともやってみてほしい。

リスト1

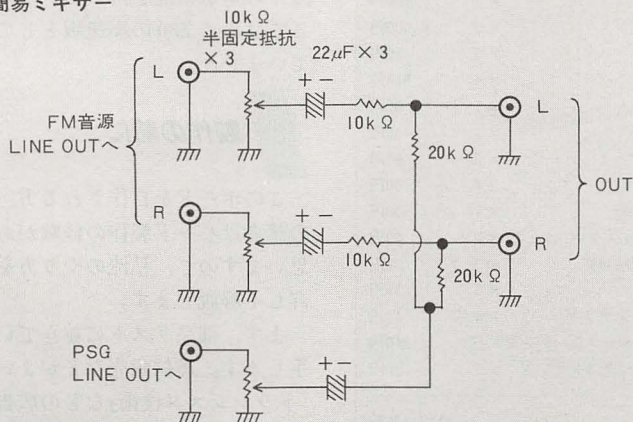
```
0000                 1 ;LIST 1
0000                 2 ; Bank subroutine call
0000                 3
0028                 4         ORG     0028H       ;1) Main RAM
0028                 5
0028 CD 2B 00        6         CALL    002BH
002B C5              7         PUSH    BC
002C F5              8         PUSH    AF
002D 3E 00           9         LD      A,0         ;Bank No.
002F 01 00 0B       10         LD      BC,0B00H
0032 ED 79          11         OUT     (C),A
0034 F1             12         POP     AF
0035 C9             13         RET
0036                14
0036                15         OFFSET  8000H
002B                16         ORG     002BH       ;2) Bank RAM
002B                17
002B 00             18         NOP
002C F5             19         PUSH    AF
002D 3E 10          20         LD      A,10H       ;Main memory
002F 01 00 0B       21         LD      BC,0B00H
0032 ED 79          22         OUT     (C),A
0034 F1             23         POP     AF
0035 C9             24         RET
```

超簡易FM/PSGミキサーの製作（X1/turbo）

西川　善司

X1のFM音源ボードを使うときの常識は「オーディオアンプにつなぐ」ということです。付属のスピーカでは低音はまったく出てきませんし，音質についてはもういいっこなしの世界です。ゲームやミュージックプログラムのなかにはPSGを使用しているものもあってFM音源はちゃんとしてても，どうもバランスがおかしくなってしまいます。「FM音源とPSGの音が一緒に出てくれたら……」と思っている人も案外多いのではないでしょうか。

そこで簡易ミキサーを作ってみました。FM音源ボードと本体から出ているAUDIO OUTをケーブルで引き出し，図のような回路で混合します。電解コンデンサ3つ（22μF），可変抵抗3つ（10kΩ），抵抗4つ（10kΩ×2，20kΩ×2），あとはコネクタだけという簡単さです。電源もなにもいりません。試作したところ，総費用380円！（ピンジャックは別）とにかく線をつなげばできあがりですから，ぜひ試してみてください。

図4　簡易ミキサー

●ハードウェア工作編

64180ボードの製作

X68000用CP/M-80システム

Yoshida Takao
吉田 孝雄

周辺ボードにもいろいろありますが，ここではいきなりCPUボードを製作します。Z80の高速版として64180を組み込み，CP/Mを超高速で動作させてみましょう。プリント基板の通販サービスもありますのでぜひ製作に挑戦してみてください。

はじめに

これから製作するのはX68000に接続する64180 CPUボードです。64180というのはX1やMZで使われている8ビットCPU，Z80の上位にあたるCPUです。今回はこのボードの製作とボード上で稼働するCP/M-80システムの制作を行います。

このシステムの特徴として，

1) 64180（10MHz）を使用
2) 64K SRAMノーリフレッシュ
3) 1ボード構成で拡張スロットへ内蔵
4) CP/MのディスクはX68000のメモリを使用。容量512Kバイト
5) Human68kとCP/Mのファイル互換プログラムを用意
6) HumanのバッチファイルでCP/Mコマンドを実行後，再びHumanに復帰し作業を継続可能

などがあります。

X68000ではCP/Mシステムのスタンダードである8インチ1Sや5インチ2Dのドライブを接続することができませんので，とりあえずソフトウェアはすべてRS-232CでCP/Mマシンから転送するということにして，ディスクにはX68000側のメモリ512KバイトをRAMディスクとして使用します。

表1 部品リスト

部品	数量	価格
日立64180-10	×1	2,000円
NEC D71055C	×1	430円
NEC D43256AC-10L	×2	6,000円
各社74LS30	×3	120円
74LS245	×4	540円
74LS365	×1	60円
74LS08	×1	40円
74LS04	×2	80円
74LS32	×4	160円
50Pユニバーサル基板	×1	4,500円
C1815 トランジスタ	×2	40円
10kΩ4P集合抵抗J48	×3	90円
10kΩ 抵抗	×7	35円
100μF 電解コンデンサ	×1	11円
0.047μF セラミックコンデンサ	×16	240円
100PF セラミックコンデンサ	×2	30円
		計12,376円

CP/MのファイルとHumanのファイルはツールにより，変換が可能です。このことは，逆にX68000用のスクリーンエディタなどの開発環境がCP/M用にも使用できるということも意味しています。高速クロック，RAMディスクとあわせて開発効率をあげることもできるでしょう。

64180とは

このボードで使っている64180という石は日立の開発したZ80上位互換のCPUで次のような特徴を持ちます。

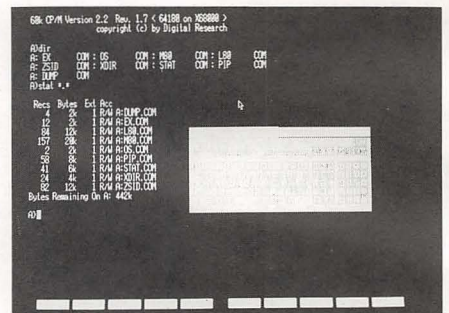
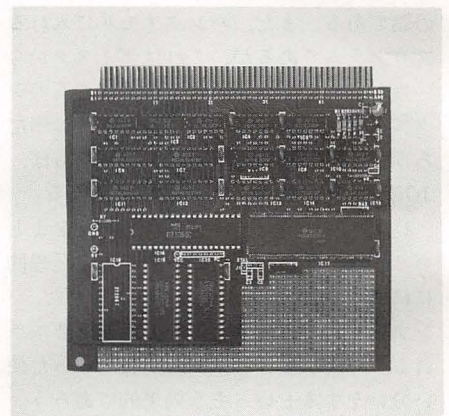
1) C-MOSで組まれており低消費電力
2) 10MHzまでの高速クロック対応
3) チップ上にMMU，DMA，シリアルI/Oなどの周辺LSIを内蔵している
4) 乗除算命令などに拡張がなされている
5) 同クロックでもZ80より2，3割処理速度が速い

今回採用した石のクロックは10MHz（ノーウェイト）ですので，このボードはコンパクトながら既存のCP/Mシステムのなかでは最高速の部類に入ります。V20やV30などを使ったシステムではi8080命令しか実行できず，CP/M以外にはほとんど使用できませんでしたが，このボードではZ80の一般的なソフトを移植することも可能です。

ただ，64180はZ80互換というだけでなく多くの追加機能を持っているのですが，ここでは単にZ80の高速版としてしか使用していません。

製作の前に

このボードを自作される方，特にOh!Xの読者はハード製作の経験があまりないと思いますので，私流のやり方を交えて少し詳しく解説します。

まず，部品リストに並んでいる部品を入手します。通信販売などがよいでしょう。『トランジスタ技術』などの広告を見て注文します。ただし，CPUのクロックやSRAMのアクセスタイムには指定のものを必ず使用してください（CPUは8MHzのものでも動作は確認しましたが)。また，コンデンサの耐電圧は16V程度のものを使ってください。抵抗の電力も1/4W以上を選ぶようにします。コンデンサ，抵抗，ICの読み方は115ページを参照してください。写真のボードではICなどはじかづけされていますが初心者の方は必ずICソケットでつけるようにしてください。

CP/Mが起動！

これが完成品だ

次に工具を揃えます。よい仕事はよい道具からです。揃えてもらいたいのは，

1) ハンダゴテ（2～30Wで先細のもの）
2) ニッパ（小型のもの）
3) ピンセット
4) ハンダ吸い取り器
5) ハンダ
6) 配線ワイヤ（ラッピング用電線。ただし，耐熱でないもの）
7) ドライバ，ラジオペンチなど一般工具

配線ワイヤは細い単線のものを使ってください。ビニル導線などは太いので配線スペースがなくなったり，X68000の狭い拡張スロットに入らなくなったりします。

製作手順

部品が揃ったところで製作に入りましょう。まず，基板に載せるICやCPUのレイアウトを決めます。回路図から考えて配線が最短になるように決めてください（とはいえ，あまり気にすることもないか)。要は正しくつながっていればそれでいいんです。

レイアウトが決まったら，部品をハンダづけしてください。おっと，基板の表裏に気をつけてください。基板エッジコネクタの番号がA1～A50と書いてあるほうが表，つまり部品面でB1～B50と書いてあるほうが裏，つまりハンダ面となります。なお，CPUは足のピッチがシュリンク(1.778mm間隔）ですのでこのままではピンの穴があいません。サンハヤトのDIP変換ソケット(基板)を入手するか，または糸鋸などで基板を切って溝を作る必要があります（私は切りました)。

続いて，実際にワイヤをハンダづけして配線を始めます。ワイヤのハンダづけにはコツがあります。必ず，図1のように行ってください。ニッパで線をむくとニッパによる傷のためワイヤがその部分から折れたり接触不良になりますのでおすすめできません。

ハンダづけはまず，ハンダしたい部分を前もってそれぞれハンダメッキしておき，次にハンダゴテでハンダを溶かしてくっつける手順でしてください。前もってハンダメッキしておかないとイモハンダやテンプラ（ハンダづけできたようでも線を引っ張ると抜ける）となって，接触不良の原因となります。

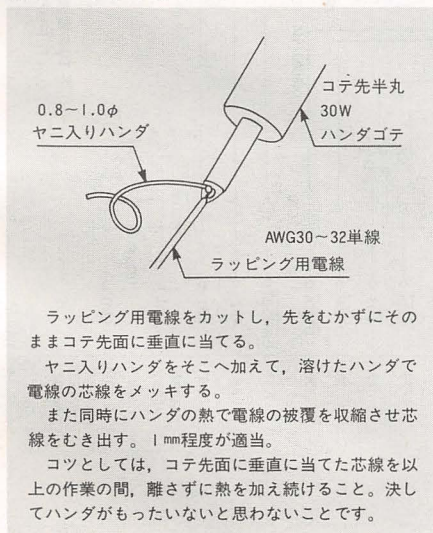
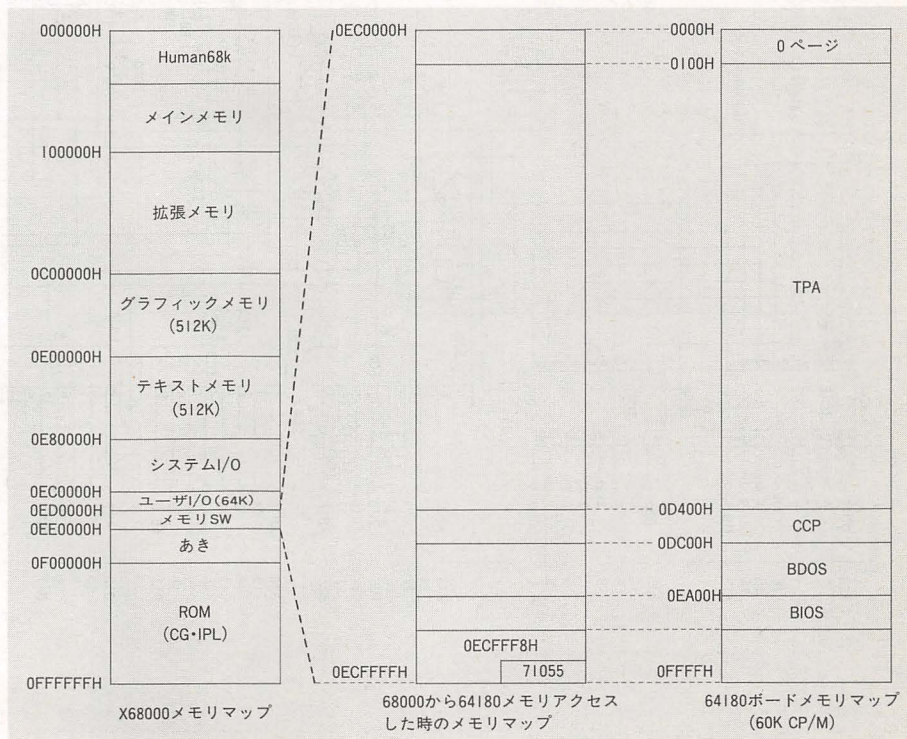



図1　ハンダづけのコツ

ラッピング用電線をカットし，先をむかずにそのままコテ先面に垂直に当てる。
ヤニ入りハンダをそこへ加えて，溶けたハンダで電線の芯線をメッキする。
また同時にハンダの熱で電線の被覆を収縮させ芯線をむき出す。1㎜程度が適当。
コツとしては，コテ先面に垂直に当てた芯線を以上の作業の間，離さずに熱を加え続けること。決してハンダがもったいないと思わないことです。

配線はICやCPUの電源から始めます。あらかじめ配線図をコピーしておいて，1本配線するごとに線を赤く塗ってチェックしてください。こうすれば，どこまでやったか忘れたりやり残すことがありません。

ここまでくればしめたもの。あとは単純作業の繰り返しです。できるだけ途中での中断は避け，腕が痛くなっても腰が文句をいっても根性でやってください。中断してかつ日があくと続ける気力が失せてくるものです。配線がすんだら，その日はもう黙って寝ること。そして翌日にはリフレッシュした頭で配線チェックをやりましょう。テスタで全配線をチェックし，特に電源＋5Vと0Vのあいだがショートしていないことを確認してください。大丈夫であれば，とりあえずICとCPUをソケットからはずしておいて，ボードのみをX68000の拡張スロットに入れてX68000を立ち上げます。大丈夫ですね？　それでは本番です。ICやCPUをソケットに入れて試してみてください。ここでHumanが立ち上がらない人はどこかに配線ミスがありますので，立ち上がるようになるまでチェックを繰り返してください。

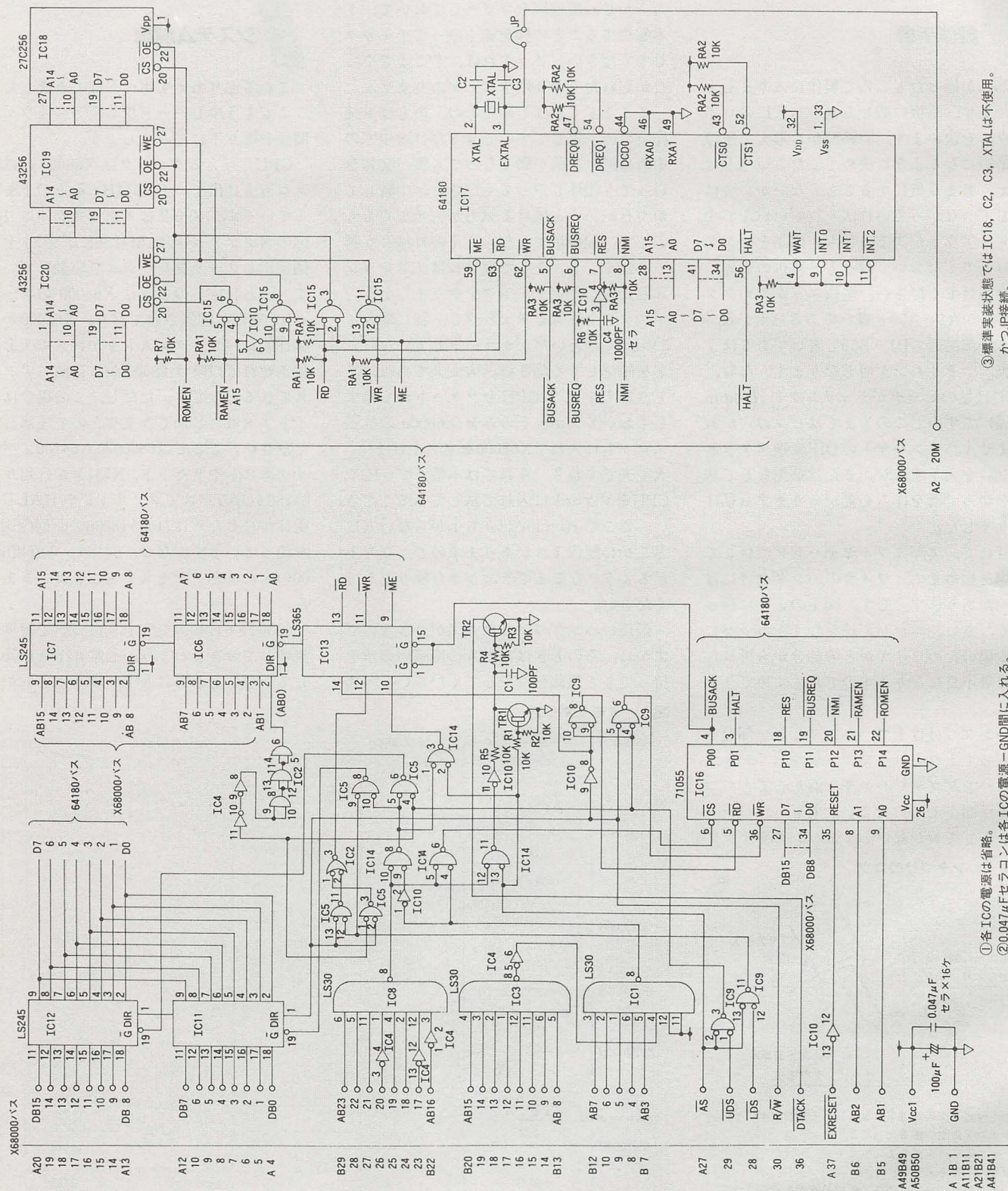
最後にハンダづけには火傷がつきものですから，そのときはただちに冷蔵庫の氷を持ってきて患部を冷やしてください。冬場であれば冷水でもいいでしょう。ハンダゴテの熱は350℃程度ですから，ほおっておいても水ぶくれができる程度ですが，一応気をつけるようにしてください。

システム解説

これでとりあえずボードはできました。ここでもう少しハードウェア/ソフトウェア関係を掘り下げてみましょう。

CPUに与えるクロックはX68000の拡張スロットに出ている20MHzをそのまま使っています。メモリはアクセスタイム100nsのスタティックRAM(256Kビット)を2個使用しノーリフレッシュで駆動しています。これは図3のようにX68000のユーザーI/O空間に配置されており，X68000から64180にバスリクエストをかけメモリを開放させれば以降は68000から自由にアクセスできるわけです。ただし，この場合はバイトアクセスしかできませんので注意してください。ここでは68000から64180にバスリクエストやリセット，NMIをかけたり，64180からのバスアクノリッジやHALTを受け付けるために71055(8255のC-MOS版)を使っています。なお，64180から直接68000のメモリをアクセスすることはできません。

68000と64180のやりとりについて簡単に説明しておきましょう。通常64180を使用していないときには64180はリセットがか

図3　メモリマップ

図3　回路図

①各ICの電源は省略。
②0.047μFセラコンは各ICの電源－GND間に入れる。
③標準実装状態ではIC18，C2，C3，XTALは不使用。
かつJP接続。

けられています。CP M.Xを実行すると64180のメモリ上にCP/Mのシステムをセットしたうえでリセットが解除され，64180が動き始めます。キー入力やコンソール出力などのBIOSコールが生じると，64180はまず0066Hからの2バイトを退避しRETNコードを書いたあとHALTします。

68000は71055経由でHALTを検知すると，バスリクエストを発生しメモリを開放させて，64180がメモリ上に書き残したメッセージを読み取ります。このメッセージは主にHuman68kに依頼するBIOSのファンクション番号とそれに付随するデータからなっています。Human はこれに従ってしかるべきDOSコールなどを行い結果を64180のメモリに書き込むわけです。

そして71055を経由してバスリクエストを解除し64180にNMIをかけます。64180はNMIにより0066Hへジャンプしますが，RETN命令により，先ほどHALTしたアドレスの直後からプログラムの実行を再開します。ここでさっき退避した0066Hからの2バイトを復帰し，BIOS コールを終了させるのです。こういった，68000の16ビットバスを8ビットバスに畳み込む方法やバスリクエストなどの使い方は京都マイクロコンピュータのTurboVシステムを参考にしています。

転送プログラムの入力

ボードができあがってもソフトがなければただの板にすぎませんから，サポートソフトを掲載します。ソフトをRS-232Cで転送することやツールなどの充実度から考えてCP/Mを乗せておくのがもっとも無難といえるでしょう。

表2を見てください。CP/Mの立ち上げに必要なファイルをまとめてあります。CP/M-80本体(CCP，BDOS，トランジェントコマンドなど）はデジタルリサーチの著作物であるため掲載することはできません。これらのソフトは8ビットパソコン用のCP/Mから抜き出してX68000にRS-232Cで転送します。よって，このシステムを立ち上げるためには，X68000とCP/MでRS-232Cが使えるパソコン，RS-232Cクロスケーブルが必要です。X68000しかない人はどこかからX1turboなどを調達してきてください。

加えて，CP/Mの使用にはライセンスが必要ですから市販のCP/M-80を購入しなければなりません。CP/Mの使用規約は1マシン1システムですから，現在手持ちの

表2　ファイル一覧表

ファイル名	説明	備考
AUTOEXEC. BAT	C：をRAMディスクとしてイニシャライズしたのち，Bドライブの RAM. BATに移行する。	Human68kシステムディスクへ入れておく。
CONFIG. SYS	グラフィックメモリをRAMディスクとして指定。	同上。
Y. DAT	RAMディスクのイニシャライズのための“Y”1文字。	同上。
CPM. S	CP/M-80用BIOSルーチンからのBIOSコールをHuman68kのDOSコールへ変換する。またCP/M呼び出しコマンドでもある。	CPM. Xのソースファイル。
RS. S	X1からのRS-232Cによるファイル転送を受信し，Human68kのファイルとして出力する。	RS. Xのソース。
HTOB. S	X1から送られたHEXファイル（RS. Xにより受信・出力されたファイル）をバイナリファイルへ復元する。	HTOB. Xのソース。
RAM. BAT	C：(RAMディスク）にCPMサブディレクトリを作り，BドライブのCP/Mファイルをコピーする。またCP/M用ディスクに指定されたファイルを転送し，CP/Mをイニシャライズして立ち上げる。	
CCPBDOS. OBJ	CP/M-80のCCP，BDOSのオブジェクト。	CPM. Xにより結合され，配置される。
BIOS. MAC	CP/M-80のBIOSでありBIOS. OBJのソース。	
EX. MAC	Human68kとCP/M-80とのファイル交換プログラム。EX. COM のソースファイル。	メディア変換プログラムではない。
OS. MAC	CP/M-80実行中にHuman68kへ戻るためのコマンド。	OS. COMのソース。
BTOH. MAC	バイナリファイルからHEXファイルを作るユーティリティ。	X1上で使用。
KEY. SYS	KEY. SYSはHuman68kオリジナルSYSファイル。	ファンクションキー定義体。
KEYCPM. SYS	KEYCPM. SYSはCP/M用。KEY. Xで作成する。	
KEY. DAT, KEYCPM. DAT	それぞれKEY. SYS, KEYCPM. SYSを実行するコマンドファイル。	KEYコマンドに対し指定。
INIT. DAT	CPMINIT. BATで実行する内容を指定する。	
M80.BAT	Human68kからCP/Mを呼び出しM80を実行。	C>M80＿ファイル名(CR)
GENCOM. BAT	同じくM80, L80を実行しCOMファイルを得る。	C>GENCOM＿ファイル名(CR)
GENBIOS. BAT	同じくM80, L80実行後BIOS. OBJとしてSYSサブディレクトリへ。	C>GENBIOS (CR)
CPM. BAT	ファンクションキーをセットしてCP/M-80へ移行。	C>CPU (CR)
CPMINIT. BAT	CP/M用ディスクを消去し，INIT. DATで指定したファイル書き込み（他のCP/Mコマンドも可）を行ってディスクをイニシャライズしたのち，CP/Mを立ち上げる。	C>CPMINIT (CR)

表3　オプション表

ファイル名	オプション	使用法	説明
CPM (Human68k)	B	C>CPM＿-B＿コマンド列 (CR)	CP/MでのBATを実行する。コマンド列にキー入力相当の命令群を入れておく。
	I	C>CPM＿-I (CR)	CP/M用ディスク領域にあてられたメモリを消去し，OS. COM，EX. COMを書き込む。また64180メモリにCP/Mシステムをロードし，CP/Mへ移行する。コールドスタートする。
	T	C>CPM＿-T (CR)	CP/M用ディスク領域のメモリを消去しない。それ以外は“I”オプションと同じ。
	なし	C>CPM (CR)	CP/M用ディスク領域にはタッチせず。またCP/Mシステムのロードはしない。(ホットスタート)
EX (CP/M)	R	A>EX＿-R＿ファイルネーム．タイプ (CR)	Human68kのファイルをCP/M用ディスクファイルとして読み出す。ワイルドカードは不可。
	W	A>EX＿-W＿ファイルネーム．タイプ (CR)	CP/MファイルをHuman68kのファイルとして書き込む。ワイルドカードは不可。
	なし	A>EX (CR)	EX. COMを起動する。以降メニューに従う。ワイルドカードは不可。
OS (CP/M)	N	A>OS＿-N (CR)	CP/MからHuman68kへ移行。BATファイル中で用いる。
	なし	A>OS (CR)	同上だが，“good bye”！と表示する。

［注意］ 1. ＿はスペース1字を示す。それ以外は不可。
2. (CR)はキャリッジリターン（↵）を示す。それ以外は不可。

CP/Mがある場合でも新たにシステムを揃えてください。転送するCP/Mは60K CP/Mであれば，PC-8801用でもFM-7 Z80ボード用でもなんでもかまいませんが，X1用のランゲージマスター（9,800円）がもっとも安価でよいでしょう。

注意点として，X1用のRS-232C マウスボード（CZ-8BM2）はX1 CP/Mでサポートされていませんので，旧式のRS-232Cボード（CZ-8RS）を使ってください（CP/Mのファイルが使える通信ソフトがあればそれを使うこと）。

X1シリーズとの組み合わせは，

X68000＋X1＋CZ-8RS＋X1 CP/M
X68000＋X1turbo＋turboCP/M＋X1 CP/M

となります。

さて，準備ができたらX1ではCP/MをX68000ではHuman68kを立ち上げます。まず，X1とX68000で通信ができるようにRS-232Cのパラメータをあわせましょう。9600 bps，8ビット，パリティなし，ストップビット2，XON/OFFなしのようにTERM.COM，SPEED.Xを使って設定します。RS-232Cボードによっては9600bpsの設定にはボード上のディップスイッチ設定が必要なものもありますので，ボードの取扱説明書を参照してください。

CP/MではPIPコマンド，HumanではCOPYコマンドによりRS-232Cへの入出力が可能ですが，

1) バイナリファイル中に1A$_H$(EOF)があると転送が中断する
2) PIPコマンドでは転送に先立って40バイトの00$_H$(NULL)を付加する
3) X1 CP/MではXON/OFF制御ができないため，X68000で受信したファイルに欠落が起きる（COPYコマンドでは受信バッファがいっぱいになるとディスクへの書き込みを行うが，この間のデータが失われる）

などのため，満足なデータ転送ができません。そこで次の3つのプログラムを用意しました。

1) BTOH.COM(CP/M)
52Kバイトまでのバイナリファイルを HEXファイルに変換する。
2) HTOB.X(Human)
HEXファイルをバイナリファイルに変換する。
3) RS.X(Human)
COPYコマンドの代わりに用いる（受信専用）。受信バッファにG-RAMを使うので欠落が起きない。

まず，BTOH.COMですが，リスト1のソースリストをWORD MASTER などのエディタを使って入力しアセンブル/リンクしてください。プログラムは MACRO-80用に記述されていますので違うアセンブラを使っている人は一部修正が必要かもしれません。以下に手順を示します(X1)。

```
A＞WM BTOH.MAC
→ソース入力
A＞M80＝BTOH
A＞L80 BTOH, BTOH/N/E
```

これで COM ファイルができました。サイズは0100$_H$～0566$_H$です。

次にHumanを立ち上げて，EDなどのエディタからHTOB.Sを打ち込みます。手順は，

```
A＞ED HTOB.S
→ソース入力
A＞AS－W HTOB
A＞LK HTOB
```

となります。これがすんだらRS.Sもまったく同様に入力してください。

転送の実際

これで転送プログラムは揃いました。今度は転送するファイルを揃えましょう。まず，CP/Mから送るファイルとして CCP，BDOSが必要なのですが，これらはシステムディスクには入っていません。しかし，CPM60N.SYSというファイルの一部がこれにあたりますのでDDTで切り取ります。

```
A＞DDT CPM60N.SYS
－M0980, 1F7F, 0100
－^C
A＞SAVE 22 CCPBDOS.OBJ
```

のような手順で行ってください。

それでは実際にファイルの転送を行ってみましょう。まず，COMファイルや SYSファイルなどのバイナリファイルはそのままでは転送できませんので先ほどの BTOH.COMでHEXファイルになおしておきます。テキストファイル（TYPEコマンドでちゃんと中身の見えるファイル）はそのままのかたちでかまいません。

CCPBDOS.OBJ を転送する場合を例に手順を示します。X1を立ち上げ，

```
A＞BTOH
Binary file name?：CCPBDOS.OBJ
Ready(Y/N)Y
```

とします。これでCCPBDOS.HEXができました。試しにTYPEすると HEX ファイルがどんなものかわかると思います。なお，HEXファイル変換は少し手抜きしたので元ファイルの約4倍の大きさのファイルを出力します。ディスクの残り容量には注意してください。

それでは転送しましょう。RS-232Cの接続と通信パラメータの設定を確認してください。X68000を立ち上げ，

```
A＞SPEED
－（そのままリターンを押す）
A＞RS
```

とします。これで受信状態になりました。SPEED.Xは通信ポートをオープンするために使っています。次にX1側から，

```
A＞PIP PUN：＝CCPBDOS.HEX
```

で送信を始めます。X68000のグラフィック画面に特徴のある模様が出れば転送は正常に行われています（G-RAMを RAMディスクに指定していると内容が破壊されます）。転送が終わるとRS.Xはファイル名を要求してきますので，

```
file name：CCPBDOS.HEX
OK?(y/n)Y
```

と入力してください。続いて HEX ファイルをバイナリファイルに戻します。X68000上で，

```
A＞HTOB
input filename：CCPBDOS.HEX
output filename：CCPBDOS.OBJ
```

のように操作します。HTOB.XもG-RAMを破壊しますので気をつけましょう。

同様に COM ファイルなど，必要なファイルを転送していきます。テキストファイルの場合はHEXファイルへの変換と復元の手間はいりません。あとはボード上のCP/MシステムのBIOSさえあれば，ひととおり揃いますね。これはリスト4のBIOS.MACをCP/M上で入力アセンブルしておく必要があります。CP/Mから，

```
A＞WM BIOS.MAC
→ソース入力
A＞M80 ＝BIOS
A＞L80 BIOS, BIOS/N/X/E
```

のように操作します。ここではBTOH.COMを使わず直接HEXファイルを出力していますが，一度バイナリファイルを作ってからBTOH.COMを使ってもかまいません。できたファイルはほかのファイルと同様にX68000に転送し，バイナリファイルに戻してください。

CP/Mの起動

これでCP/Mのシステム一式が転送されました。これをボード上に転送して64180に制御を移せばCP/Mシステムのできあがりですが，そのために必要なのがプログラ

ムがCPM.Xというファイルです。これもHTOB.XやRS.Xと同様にソースを打ち込んでアセンブル/リンクしてください。

図4は立ち上げ時に推奨されるファイル配置図です。バッチファイルやデータなどはリストどおりにエディタでファイルを作って，図4のようなファイル構成のディスクを作っていってください。

立ち上げはここで作った2枚のディスクをドライブに入れ，ハードディスクなどは接続しない状態でリセットすればRAMディスクを初期化し，CP/MのソフトをRAMディスクに転送してオートスタートします。この間約37秒です。OS.COMを実行すればHumanに戻りますし，コマンドモードからCPMと入力すればCP/Mを起動できます。

操作方法は通常のCP/Mシステムと同じですが，注意点を含めて説明しておきます。まず，CP/MにはX68000の特徴である日本語入出力，グラフィック，サウンドなどはサポートされていません。もちろんBIOSコールを拡張し，HumanのDOSコールと接続すれば比較的簡単にかなりの機能を引き出せるのですが，こういったものはCP/Mで使うべきものではなく68000の機械語なり高級言語なりから使うのが正当というものでしょう。

ファイルのなかにCPMINIT.BATというものがありますね。AUTOEXEC.BATは起動するとBドライブの¥CPMディレクトリに含まれているファイルのうち，INIT.DAT内に記述されているファイルをCP/M用のデバイスに転送しますので皆さんの好みに従って書き換えてください。なお，このプログラムに限らず，CP/Mで実行するバッチファイルでは@をリターンキーの代わりに使用し，(CP/Mの都合で)ダミーのキー入力が必要ですから@の後ろにはスペースを1字分置いています。また，CP/Mではなるべく大文字を使用するようにしてください。

全体的な操作のポイントとしてエディタなどはHumanのものを使い，プログラムの実行だけバッチファイルでCP/Mを起動して自動復帰させることをおすすめします。ファンクションキーを用意していますが，やはりCP/Mの操作性はHumanより劣ります。その点，バッチファイルであれば操作性は非常に良好です。

また，CP/Mの使用中に64180が暴走してしまった場合はX68000のインタラプトスイッチを押してください。CP/Mのコマンド待ちのときには^Cを押しっぱなしにして

図4　ファイルの配置図

表4　バッチファイルなど

```
●AUTOEXEC.BAT
ECHO␣OFF(CR)
VERIFY␣ON(CR)
BREAK␣ON(CR)
PATH␣¥;¥BIN;¥BASIC;¥福袋(CR)
FORMAT␣C:␣/C␣<Y.DAT␣>NUL(CR)
B:RAM(EOF)
●CONFIG.SYS
FILES(TAB)=␣15(CR)
BUFFERS(TAB)=␣20(CR)
BELL(TAB)=␣BEEP.SYS(CR)
DEVICE(TAB)=␣RAMDISK.SYS␣#G(CR)
DEVICE(TAB)=␣PRNDRV.SYS(CR)
DEVICE(TAB)=␣PCMDRV.SYS(CR)
VERIFY(TAB)=␣ON
(EOF)
●Y.DAT
Y(EOF)      ㊟Yの1文字のみ
●INIT.DAT
EX␣-R␣M80.COM(CR)
EX␣-R␣L80.COM(CR)
EX␣-R␣ZSID.COM(CR)
EX␣-R␣STAT.COM(CR)
EX␣-R␣PIP.COM(CR)
EX␣-R␣DUMP.COM(CR)
OS␣-N(CR)
(EFO)
●GENCOM.BAT
ECHO␣OFF(CR)
¥CPM¥SYS¥CPM.X␣-B␣EX␣-R␣%1.MAC@␣M80␣=%1@␣L80␣%1,%1/N/E@␣EX␣-W␣%1.
COM@␣OS␣-N@(EOF)
●GENBIOS.BAT
ECHO␣OFF(CR)
¥CPM¥SYS¥CPM.X␣-B␣EX␣-R␣BIOS.MAC@␣M80␣=BIOS@␣L80␣BIOS,BIOS.OBJ/N/E@␣EX␣
-W␣BIOS.OBJ@␣OS␣-N@(CR)
COPY␣¥CPM¥BIOS.OBJ␣¥CPM¥SYS(CR)
DEL␣¥CPM¥BIOS.OBJ(EOF)
●CPM.BAT
ECHO␣OFF(CR)
KEY.X␣<¥CPM¥SYS¥KEYCPM.DAT␣>¥CPM¥SYS¥TRASH.DAT(CR)
¥CPM¥SYS¥CPM.X␣%1(CR)
KEY.X␣<¥CPM¥SYS¥KEY.DAT␣>¥CPM¥SYS¥TRASH.DAT(CR)
DEL␣¥CPM¥SYS¥TRASH.DAT(EOF)
●CPMINIT.BAT
ECHO␣OFF(CR)
¥CPM¥SYS¥CPM.X␣-I␣<¥CPM¥SYS¥INIT.DAT␣>NUL(CR)
CPM␣-T(EOF)

●RAM.BAT
ECHO␣OFF(CR)
C:(CR)
MD␣CPM(CR)
CD␣CPM(CR)
MD␣SYS(CR)
COPY␣B:¥CPM¥SYS␣SYS␣>NUL(CR)
MD␣BAT(CR)
COPY␣B:¥CPM¥BAT␣BAT␣>NUL(CR)
COPY␣B:¥CPM␣>NUL(CR)
PATH␣C:¥CPM¥BAT;A:¥;A:¥BIN;A:¥福袋(CR)
¥CPM¥SYS¥CPM.X␣-I␣<¥CPM¥SYS¥INIT.DAT␣>NUL
(CR)
CPM␣-T(EOF)
●KEY.DAT
L¥CPM¥SYS¥KEY.SYS(CR)
(EOF)
●KEY CPM.DAT
L¥CPM¥SYS¥KEYCPM.SYS(CR)
(EOF)
●M80.BAT
ECHO␣OFF(CR)
¥CPM¥SYS¥CPM.X␣-B␣EX␣-R␣%1.MAC@␣M80
␣=%1@␣OS␣-N@(EOF)
```

もHumanに復帰できます。

さらに注意点として以下のものがあります。

1)　ファイル関係

CP/M用のデバイスとしては,メインメモリのうち512Kバイトを非常駐のかたちで確保しており,以後はこれをRAMディスクとして使用します。X68000のディスクドライブは使用しませんので気をつけてください。1ドライブ仕様ですが容量が512Kバイトありますので通常の使用状況ではほとんど問題ないでしょう。不満な方はBIOS.MACおよびCPM.Sを変更しRAMディスクを拡張するか,ディスク対応に変更してください。また,使用後,必要なファイルはEX.COMを使ってHumanに転送するようにしてください。

2)　コンソール入出力

入出力はHuman68kのDOSコールを使用しています。なお,エスケープシーケンスなどはX1のものと異なっていますので標準装備のWORD MASTERなど,コンフィグレーションのできないソフトは使用できません。

3)　その他の入出力

RS-232CについてはHumanのIOCSコールを使ってX68000のRS-232Cポートをそのまま使用できます。DOSコールを使わなかった理由は,Human内での処理のため一部のコードで通信がストップすることがあるからです。

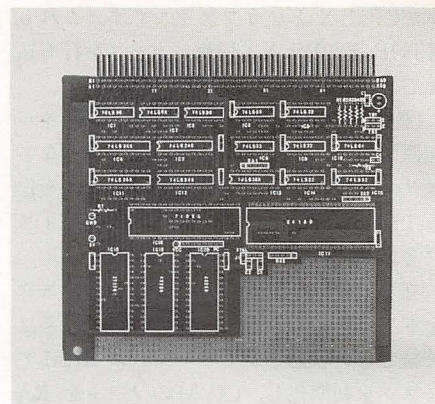
プリンタについてはDOSコールを使いました。もちろんCP/M実行中に^P(コントロールP)などでプリントアウトできますし,PIPコマンドでLST:(LPT:)として使用できます。

今後の展開

今回はとりあえず,CP/M-80を移植しましたが,X1時代のCP/Mの速度,操作性に対し比較にならないほどの性能アップがはかれました。ただし,残念ながらCP/MはやはりCP/MであってHumanのアプリケーションではないため,ファイルが二重になったりヒストリなどの便利な機能が使えなかったりといった弱点もあります。

これを根本的に解決するにはHuman上で動くCP/Mエミュレータを作成し,CP/Mプログラムをひとつのアプリケーションとして実行させることが考えられます。HumanではMS-DOSにあったCP/MコンパチのBDOSコールがないため,私の力では実現できませんでした。今後の課題といえるでしょう。

また,Z80コンパチであれば当然,S-OS“SWORD”などの移植も考えられます。かつて,Huモニタコンパチのモニタを作ってX1用“SWORD”からPC-8801オールRAM“SWORD”を作った例もありますが,それと同じ方法がもっとも簡単そうです。

いずれにしても,現状では2個のCPUが並列処理を行うことはありませんので,いまひとつ面白味に欠けます。これらを同時に動かすための64180ドライバを開発して組み込むことが重要でしょう。CP/Mと“SWORD”が走れば64180ボード用プログラムの開発もずっとはかどるはずです。

実装前のプリント基板

最後になりましたが,配線のスパゲティになった基板では心もとない,かつ暇はないけど金ならあるぞという方のために私が製作したプリント基板を実費でお分けできることになりました。

1)　プリント基板のみ(5千円前後の予定)

2)　プリント基板+64180+71055+RAM

自分の住所を書き,60円切手を貼った返信用封筒を同封し,1),2)のいずれかを明記して,Oh!X編集室「64180プリント基板」係まで封書で申し込んでください。整理の都合上ひとまず,1カ月間(1月18日到着分まで)で締め切らせていただき,正確な価格などは追って連絡いたします。

Profile

◇吉田さんは神奈川県にお住まいの26歳,技術系の会社員です。X1を経て現在X68000ユーザー。CP/Mはいつも仕事関係で必要なため,このボードを作ったそうです。

リスト1　BTOH.MAC

```
==================  BTOH.MAC  ==================
     1: ; ****************************************
     2: ; *                                      *
     3: ; * program name : BTOH.MAC              *
     4: ; *                                      *
     5: ; * CONVERT BINARY FILE TO HEX TYPE OBJECT *
     6: ; *                                      *
     7: ; *    1988  9/17  BY TK-YSD             *
     8: ; *                                      *
     9: ; ****************************************
    10: ;
    11: ;
    12:         .Z80
    13: ;
    14:         ASEG
    15: ;
    16: ;
    17: PBEGIN  EQU     0D200H          ; PROGRAM BEGIN ADDRESS.
    18: BINBUF  EQU     100H            ; BINARY  BUFF. ADDRESS.
    19: HEXBUF  EQU     0D100H          ; HEX     BUFF. ADDRESS.
    20: ;
    21: BDOS    EQU     5
    22: ;
    23: ;
    24:         ORG     100H
    25: ;
    26: ;
    27: ; BLOCK TRANSFER AND JUMP TO MAIN ROUTINE
    28: ;
    29:         LD      HL,PSTART
    30:         LD      DE,MAIN
    31:         LD      BC,PSTOP-PSTART
    32:         LDIR
    33:         JP      MAIN
    34: ;
    35: PSTART:
    36: ;
    37:         .PHASE  PBEGIN
    38: ;
    39: ; MAIN ROUTINE
    40: ;
    41: MAIN:
    42:         LD      BC,HEXBUF-BINBUF-1      ; CLEAR BUFFER.
    43:         LD      DE,BINBUF+1
    44:         LD      HL,BINBUF
    45:         LD      (HL),0
    46:         LDIR
    47: ;
    48:         LD      DE,MESS                 ; START MESSAGE.
    49:         CALL    MESSAG
    50: ;
    51: MAIN0:
    52:         LD      DE,DREAD
    53:         CALL    MESSAG
    54:         CALL    FILNAM                  ; READ FILE NAME.
    55:         AND     A
    56:         JR      NZ,MAIN0
    57: ;
    58:         CALL    FCB2ST
    59:         LD      DE,DSAVE
    60:         CALL    MESSAG
    61:         LD      DE,DREADY
    62:         CALL    MESSAG
    63:         CALL    KEY
    64:         CP      'Y'
    65:         RET     NZ
    66: ;
    67:         CALL    READ                    ; READ BIN FILE.
    68:         CALL    SAVE                    ; SAVE HEX FILE.
    69:         AND     A
    70:         RET     NZ
    71: ;
    72:         CALL    OK
    73:         JP      0                       ; RETURN TO CP/M.
    74: ;
    75: ;
    76: ; READ FILE
    77: ;
    78: READ:
    79:         CALL    OPEN
    80:         INC     A
    81:         JP      Z,OPNERR
    82: ;
    83:         LD      DE,BINBUF
    84: READ0:
    85:         CALL    READSQ
    86:         OR      A
    87:         RET     NZ
    88: ;
    89:         LD      HL,80H
    90:         LD      BC,80H
    91:         LDIR
    92:         LD      (ENDADR),DE
    93:         JR      READ0
    94: ;
    95: ; SAVE FILE
    96: ;
    97: SAVE:
    98:         CALL    DELETE
    99:         CALL    MAKE
   100:         INC     A
   101:         JP      Z,DISKER
   102: ;
   103:         LD      HL,BINBUF               ; CONV START ADDRESS.
```

```
104:         LD      (LODADR),HL
105: SAVE0:
106:         CALL    CONV
107:         LD      BC,80H
108:         LD      DE,80H
109:         LD      HL,HEXBUF
110:         LDIR
111: ;
112:         CALL    WRITSQ
113:         OR      A
114:         JP      NZ,DISKER
115: ;
116:         LD      DE,(ENDADR)
117:         LD      HL,(LODADR)
118:         AND     A
119:         SBC     HL,DE
120:         JR      C,SAVE0
121: ;
122:         LD      BC,16
123:         LD      DE,80H
124:         LD      HL,HEXEND
125:         LDIR
126: ;
127:         CALL    WRITSQ
128:         OR      A
129:         JP      NZ,DISKER
130: ;
131:         CALL    CLOSE
132:         XOR     A                       ; OK MARK.
133:         RET
134: ;
135: ; CONVERT
136: ;
137: CONV:
138:         LD      B,2
139:         LD      HL,HEXBUF               ; SAVE START ADDRESS.
140: CONV0:
141:         LD      (HL),':'                ; REC MARK.
142:         INC     HL
143: ;
144:         LD      (HL),'1'                ; REC LEN.
145:         INC     HL
146:         LD      (HL),'0'
147:         INC     HL
148: ;
149:         PUSH    BC                      ; LOAD ADR.
150:         LD      BC,(LODADR)
151:         LD      A,16                    ; CHK SUM INIT.
152:         ADD     A,B
153:         ADD     A,C
154:         LD      (CHKSUM),A
155:         LD      A,B
156:         CALL    ASCII
157:         LD      (HL),D
158:         INC     HL
159:         LD      (HL),E
160:         INC     HL
161: ;
162:         LD      A,C
163:         CALL    ASCII
164:         LD      (HL),D
165:         INC     HL
166:         LD      (HL),E
167:         INC     HL
168:         POP     BC
169: ;
170:         LD      (HL),'0'                ; REC TYP.
171:         INC     HL
172:         LD      (HL),'0'
173:         INC     HL
174: ;
175:         LD      IX,(LODADR)             ; DATA.
176:         LD      A,16
177:         LD      C,0                     ; CHK SUM.
178: CONV1:
179:         PUSH    AF
180:         LD      A,(IX+0)
181:         INC     IX
182: ;
183:         PUSH    AF
184:         ADD     A,C
185:         LD      C,A
186:         POP     AF
187: ;
188:         CALL    ASCII
189:         LD      (HL),D
190:         INC     HL
191:         LD      (HL),E
192:         INC     HL
193: ;
194:         POP     AF
195:         DEC     A
196:         JR      NZ,CONV1
197: ;
198:         LD      (LODADR),IX
199: ;
200:         LD      A,(CHKSUM)              ; CHK SUM.
201:         ADD     A,C
202:         NEG
203:         CALL    ASCII
204:         LD      (HL),D
205:         INC     HL
206:         LD      (HL),E
207:         INC     HL
208: ;
209:         LD      (HL),0DH
210:         INC     HL
211:         LD      (HL),0AH
212:         INC     HL
213:
214:         LD      A,19
215: CONV2:
216:         LD      (HL),0
217:         INC     HL
218:         DEC     A
219:         JR      NZ,CONV2
220:
221:         DJNZ    CONV0                   ; 128 BYTES?
222: ;
223:         RET
224: ;
225: ; FILE OPEN
226: ;
227: OPEN:
228:         PUSH    BC
229:         PUSH    DE
230:         PUSH    HL
231:         LD      C,15
232:         LD      DE,FCB
233:         CALL    BDOS
234:         POP     HL
235:         POP     DE
236:         POP     BC
237:         RET
238: ;
239: ; CLOSE FILE
240: ;
241: CLOSE:
242:         PUSH    BC
243:         PUSH    DE
244:         PUSH    HL
245:         LD      C,16
246:         LD      DE,FCB2
247:         CALL    BDOS
248:         POP     HL
249:         POP     DE
250:         POP     BC
251:         RET
252: ;
253: ; DELETE FILE
254: ;
255: DELETE:
256:         PUSH    BC
257:         PUSH    DE
258:         PUSH    HL
259:         LD      C,19
260:         LD      DE,FCB2
261:         CALL    BDOS
262:         POP     HL
263:         POP     DE
264:         POP     BC
265:         RET
266: ;
267: ; READ SEQUENTIAL FILE
268: ;
269: READSQ:
270:         PUSH    BC
271:         PUSH    DE
272:         PUSH    HL
273:         LD      C,20
274:         LD      DE,FCB
275:         CALL    BDOS
276:         POP     HL
277:         POP     DE
278:         POP     BC
279:         RET
280: ;
281: ; WRITE SEQUENTIAL FILE
282: ;
283: WRITSQ:
284:         PUSH    BC
285:         PUSH    DE
286:         PUSH    HL
287:         LD      C,21
288:         LD      DE,FCB2
289:         CALL    BDOS
290:         POP     HL
291:         POP     DE
292:         POP     BC
293:         RET
294: ;
295: ; MAKE FILE
296: ;
297: MAKE:
298:         PUSH    BC
299:         PUSH    DE
300:         PUSH    HL
301:         LD      C,22
302:         LD      DE,FCB2
303:         CALL    BDOS
304:         POP     HL
305:         POP     DE
306:         POP     BC
307:         RET
308: ;
309: ; GET BYTE DATA FROM 2 BYTES ASCII
310: ;
311: BYTE:
312:         LD      A,D
313:         CALL    SBYTE
314:         CALL    LSHIFT
315:         LD      D,A
316:         LD      A,E
317:         CALL    SBYTE
318:         ADD     A,D
319:         RET
320: ;
321: ; SUB BYTE
322: ;
323: SBYTE:
324:         SUB     30H
325:         CP      10
326:         RET     C
327:         SUB     7
328:         RET
329: ;
330: ; GET 2BYTES ASCII FROM BYTE DATA
331: ;
332: ASCII:
333:         PUSH    AF
334:         AND     0F0H
335:         CALL    RSHIFT
336:         CALL    SASCII
337:         LD      D,A
338:         POP     AF
339:         AND     0FH
340:         CALL    SASCII
341:         LD      E,A
342:         RET
343: ;
344: ; SUB ASCII
345: ;
346: SASCII:
347:         ADD     A,30H
348:         CP      3AH
349:         RET     C
350:         ADD     A,7
351:         RET
352: ;
353: OK:
354:         LD      DE,DOK
355:         CALL    MESSAG
356:         RET
357: ;
358: ; ASCII...A
359: ;
360: ASCA:
361:         PUSH    AF
362:         CALL    RSHIFT
363:         CALL    EXIT
364:         POP     AF
365:         CALL    EXIT
366:         RET
367: ;
368: EXIT:
369:         AND     0FH
370:         CP      0AH
371:         JR      C,EXIT0
372:         ADD     A,7
373: EXIT0:
374:         ADD     A,30H
375:         CALL    TYPE
376:         RET
377: ;
378: ; ASCII...HL
379: ;
380: ASCHL:
381:         LD      A,H
382:         CALL    ASCA
383:         LD      A,L
384:         CALL    ASCA
385:         CALL    SPACE
386:         RET
387: ;
388: ; SPACE
389: ;
390: SPACE:
391:         LD      A,' '
392:         CALL    TYPE
393:         RET
394: ;
395: ; CR,LF
396: ;
397: CRLF:
398:         LD      A,0DH
399:         CALL    TYPE
```

```
400:         LD      A,0AH
401:         CALL    TYPE
402:         RET
403: ;
404: ; GET DECIMAL NUMBER
405: ;
406: ;
407: DECIML:
408:         PUSH    BC
409:         PUSH    DE
410:         LD      B,4                     ; MAX 4 TIMES.
411:         LD      HL,0
412: DECML0:
413:         CALL    DECBYT
414:         CP      0FFH                    ; END?
415:         JR      Z,DECML1
416:         LD      D,H
417:         LD      E,L
418:         ADD     HL,HL
419:         ADD     HL,HL
420:         ADD     HL,DE
421:         ADD     HL,HL                   ; HL=HL*10.
422:         LD      D,0
423:         LD      E,A
424:         ADD     HL,DE
425:         DJNZ    DECML0
426: DECML1:
427:         POP     DE
428:         POP     BC
429:         RET
430: ;
431: ; GET HEXADECIMAL NUMBER
432: ;
433: ;
434: HEXDEC:
435:         PUSH    BC
436:         PUSH    DE
437:         LD      B,4                     ; MAX 4 TIMES.
438:         LD      HL,0
439: HEXDC0:
440:         CALL    HEXBYT
441:         CP      0FFH                    ; END?
442:         JR      Z,HEXDC1
443:         ADD     HL,HL
444:         ADD     HL,HL
445:         ADD     HL,HL
446:         ADD     HL,HL                   ; HL=HL*16.
447:         LD      D,0
448:         LD      E,A
449:         ADD     HL,DE
450:         DJNZ    HEXDC0
451: HEXDC1:
452:         POP     DE
453:         POP     BC
454:         RET
455: ;
456: ; KEY IN DECIMAL NUMBER
457: ;
458: ;  EXIT...ERROR=0FFH
459: ;
460: DECBYT:
461:         CALL    KEY
462:         SUB     30H
463:         JR      C,DECERR
464:         CP      10
465:         RET     C                       ; 0-9 OK.
466: DECERR:
467:         LD      A,0FFH                  ; ERROR MARK.
468:         RET
469: ;
470: ; KEY IN HEXADECIMAL NUMBER
471: ;
472: ;  EXIT...ERROR=0FFH
473: ;
474: HEXBYT:
475:         CALL    KEY
476:         SUB     30H
477:         JR      C,HEXERR
478:         CP      10
479:         RET     C                       ; 0-9 OK.
480:         SUB     7
481:         CP      10
482:         JR      C,HEXERR
483:         CP      16
484:         RET     C                       ; 10-15 OK.
485: HEXERR:
486:         LD      A,0FFH                  ; ERROR MARK.
487:         RET
488: ;
489: ; SHIFT RIGHT 4 BITS
490: ;
491: RSHIFT:
492:         RRCA
493:         RRCA
494:         RRCA
495:         RRCA
496:         RET
497: ;
498: ; SHIFT LEFT 4 BITS
499: ;
500: LSHIFT:
501:         RLCA
502:         RLCA
503:         RLCA
504:         RLCA
505:         RET
506: ;
507: ; CONSOLE INPUT
508: ;
509: KEY:
510:         PUSH    BC
511:         PUSH    DE
512:         PUSH    HL
513:         LD      C,1
514:         CALL    BDOS
515:         POP     HL
516:         POP     DE
517:         POP     BC
518:         RET
519: ;
520: ; CONSOLE OUTPUT
521: ;
522: TYPE:
523:         PUSH    BC
524:         PUSH    DE
525:         PUSH    HL
526:         LD      C,2
527:         LD      E,A
528:         CALL    BDOS
529:         POP     HL
530:         POP     DE
531:         POP     BC
532:         RET
533: ;
534: ; PRINT STRING
535: ;
536: MESSAG:
537:         PUSH    BC
538:         PUSH    DE
539:         PUSH    HL
540:         LD      C,9
541:         CALL    BDOS
542:         POP     HL
543:         POP     DE
544:         POP     BC
545:         RET
546: ;
547: ; READ CONSOLE BUFFER
548: ;
549: GETLIN:
550:         PUSH    BC
551:         PUSH    DE
552:         PUSH    HL
553: ; CLEAR FILBUF
554:         LD      B,40
555:         LD      HL,FILBUF
556: GTLIN0:
557:         LD      (HL),0
558:         INC     HL
559:         DJNZ    GTLIN0
560: ;
561:         LD      A,38                    ; MAX 38 CHARACTER.
562:         LD      (FILBUF),A
563:         LD      C,10
564:         LD      DE,FILBUF
565:         CALL    BDOS
566:         POP     HL
567:         POP     DE
568:         POP     BC
569:         RET
570: ;
571: ; GET FILE NAME FROM CONSOLE
572: ;
573: FILNAM:
574:         CALL    GETLIN
575: ; CLEAR FCB
576:         LD      B,40
577:         LD      HL,FCB
578: FILNM0:
579:         LD      (HL),0
580:         INC     HL
581:         DJNZ    FILNM0
582: ; FILE NAME CHECK
583:         LD      B,8
584:         LD      DE,FCB+1
585:         LD      HL,FILBUF+2
586: FILNM1:
587:         LD      A,(HL)
588:         INC     HL
589:         CP      '.'
590:         JR      Z,FILNM2
591:         LD      (DE),A
592:         INC     DE
593:         DJNZ    FILNM1
594:         LD      A,(HL)
595:         INC     HL
596:         CP      '.'
597:         JR      NZ,FNERR
598:         JR      FILNM4
599: FILNM2:
600:         LD      A,20H
601: FILNM3:
602:         LD      (DE),A
603:         INC     DE
604:         DJNZ    FILNM3
605: FILNM4:
606:         LD      B,3
607: FILNM5:
608:         LD      A,(HL)
609:         INC     HL
610:         CP      0DH                     ; END?
611:         RET     Z
612:         LD      (DE),A
613:         INC     DE
614:         DJNZ    FILNM5
615:         XOR     A                       ; OK MARK.
616:         RET
617: ;
618: ; MAKE FCB2
619: ;
620: FCB2ST:
621:         LD      BC,40
622:         LD      DE,FCB2
623:         LD      HL,FCB
624:         LDIR
625:         LD      HL,FCB2+9               ; FILE TYPE.
626:         LD      (HL),'H'
627:         INC     HL
628:         LD      (HL),'E'
629:         INC     HL
630:         LD      (HL),'X'
631:         RET
632: ;
633: ; OPEN ERROR
634: ;
635: OPNERR:
636:         LD      DE,DOPEN
637:         CALL    MESSAG
638:         JP      0
639: ;
640: FNERR:
641:         LD      DE,DFNERR
642:         CALL    MESSAG
643:         JP      0
644: ;
645: DISKER:
646:         LD      DE,DERROR
647:         CALL    MESSAG
648:         CALL    DELETE
649:         JP      0
650: ;
651: DREADY:
652:         DEFB    0DH,0AH
653:         DEFB    '  Ready? (Y/N) $'
654: ;
655: DREAD:
656:         DEFB    0DH,0AH
657:         DEFB    '  Binary file name? : $'
658: ;
659: DSAVE:
660:         DEFB    0DH,0AH
661:         DEFB    '  Save name is  filename.hex'
662:         DEFB    0DH,0AH,'$'
663: ;
664: DOK:
665:         DEFB    0DH,0AH
666:         DEFB    0DH,0AH
667:         DEFB    '   Complete'
668:         DEFB    0DH,0AH,'$'
669: ;
670: DOPEN:
671:         DEFB    0DH,0AH
672:         DEFB    0DH,0AH
673:         DEFB    '   No file'
674:         DEFB    0DH,0AH,'$'
675: ;
676: DERROR:
677:         DEFB    0DH,0AH
678:         DEFB    0DH,0AH
679:         DEFB    '  Disk error'
680:         DEFB    0DH,0AH,'$'
681: ;
682: DFNERR:
683:         DEFB    0DH,0AH
684:         DEFB    0DH,0AH
685:         DEFB    '   File name error'
686:         DEFB    0DH,0AH,'$'
687: ;
688: MESS:
689:         DEFB    0DH,0AH
690:         DEFB    ' Utility for Binary File to HEX type object converter'
691:         DEFB    0DH,0AH,'$'
692: ;
693: HEXEND:
694:         DEFB    ':00000001FF',1AH,1AH,1AH,1AH,1AH
695: ;
```

▶12月号の「われら電脳遊戯民」を読んで，『ノーライフキング』を買ってしまった。この本はとても面白くて，時間の経つのも忘れてしまうほどでした。みんなもこの『ノーライフキング』を読んだらいいなぁ，と思います。　宮島　英礼 (18) 神奈川県

```
696: ;
697: ; WORK AREA
698: ;
699: ;
700: CHKSUM:
701:         DEFS    1
702: ;
703: LODADR:
704:         DEFS    2
705: ;
706: ENDADR:
707:         DEFS    2
708: ;
709: FCB:
710:         DEFS    40
711: ;
712: FCB2:
713:         DEFS    40
714: ;
715: FILBUF:
716:         DEFS    40
717: ;
718: ;
719:         .DEPHASE
720: ;
721: PSTOP:
722: ;
723: ;
724:         END
```

リスト2 HTOB.S

```
================== HTOB.S ==================
    1: ****************************
    2: *                          *
    3: * program name : HTOB.S    *
    4: *                          *
    5: * HEX to Binary converter  *
    6: *                          *
    7: *   1988 9/16  by TK-YSD   *
    8: *                          *
    9: ****************************
   10:
   11: hex_buff        equ     $e40000         ; TEXT VRAM address.
   12: bin_buff        equ     hex_buff+$010000
   13:
   14:
   15:         .text
   16:
   17:
   18: **********************
   19: *      main          *
   20: **********************
   21: main:
   22:         clr.l   -(sp)
   23:         dc.w    _super                  ; system mode.
   24:         move.l  d0,sp
   25:
   26:         bsr     rdfile                  ; load HEX file.
   27:         bsr     conv                    ; HEX to Binary.
   28:         bsr     wrfile                  ; save Bin file.
   29:
   30:         move.l  #hex_buff,a0            ; clear memory.
   31: main0:
   32:         move.l  #0,(a0)+
   33:         cmp.l   #hex_buff+$40000,a0
   34:         bne     main0
   35:
   36:         bsr     crlf
   37:         bsr     crlf
   38:         jmp     exit                    ; return to OS.
   39:
   40:
   41: **********************
   42: *     rdfile         *
   43: **********************
   44: rdfile:
   45:         bsr     rdfsub
   46: rdfil0:
   47:         move.l  adr180,a0               ; data pointer.
   48:         move.w  fileno0,d0              ; file handle number.
   49:         move.l  #65536,d1               ; read file size.
   50:         bsr     read
   51:         tst.l   d0
   52:         bmi     rdfil3                  ; jump, if error.
   53:
   54:         beq     rdfil1                  ; jump, if file end.
   55:
   56:         add.l   d0,adr180
   57:         bra     rdfil0
   58:
   59: rdfil1:
   60:         move.w  fileno0,d0              ; read file handle number.
   61:         bsr     close
   62:         tst.l   d0
   63:         bmi     rdfil5
   64:
   65: rdfil2:
   66:         bsr     crlf
   67:         rts
   68:
   69: rdfil3:
   70:         move.l  #remsg,prnadr
   71:         bsr     print
   72:         jmp     exit                    ; return to OS.
   73:
   74: rdfil4:
   75:         move.l  #nfmsg,prnadr
   76:         bsr     print
   77:         jmp     exit                    ; return to OS.
   78:
   79: rdfil5:
   80:         move.l  #clmsg,prnadr
   81:         bsr     print
   82:         jmp     exit                    ; return to OS.
   83:
   84:
   85: **********************
   86: *     rdfsub         *
   87: **********************
   88: rdfsub:
   89:         move.l  #name0,a0               ; name pointer.
   90:         move.l  #hfnmsg,prnadr
   91:         bsr     fname                   ; get file name.
   92:
   93:         move.l  #hex_buff,adr180
   94:
   95:         move.l  #name0,a0               ; name pointer.
   96:         clr.w   d0                      ; read file mode.
   97:         bsr     open
   98:         tst.l   d0
   99:         bmi     rdfil4                  ; jump, if error.
  100:
  101:         move.w  d0,fileno0              ; file handle number.
  102:         rts
  103:
  104:
  105: **********************
  106: *      conv          *
  107: **********************
  108: conv:
  109:         move.l  #hex_buff,a0
  110:         move.l  #bin_buff,a1
  111: conv0:
  112:         cmp.b   #':',(a0)+
  113:         bne     conv0
  114:
  115:         bsr     atob                    ; get record length.
  116:         move.b  d0,count
  117:         move.b  d0,chksum               ; init. check sum.
  118:
  119:         bsr     atob                    ; get load address.
  120:         add.b   d0,chksum
  121:         bsr     atob
  122:         add.b   d0,chksum
  123:
  124:         bsr     atob                    ; get record type.
  125:         cmp.b   #1,d0
  126:         beq     conv2                   ; jump, if record end.
  127:
  128: conv1:
  129:         bsr     atob                    ; get data.
  130:         add.b   d0,chksum
  131:         move.b  d0,(a1)+
  132:         subq.b  #1,count
  133:         bne     conv1
  134:
  135:         bsr     atob                    ; get check sum.
  136:         add.b   d0,chksum
  137:         tst.b   chksum
  138:         beq     conv0
  139:
  140:         move.l  #hexmsg,prnadr
  141:         bsr     print
  142:         jmp     exit                    ; return to OS.
  143:
  144: conv2:
  145:         move.l  a1,end_adrs
  146:         rts
  147:
  148:
  149: **********************
  150: *     wrfile         *
  151: **********************
  152: wrfile:
  153:         move.l  #name1,a0               ; name pointer.
  154:         move.l  #bfnmsg,prnadr
  155:         bsr     fname                   ; get file name.
  156:
  157:         move.l  #bin_buff,d4            ; start address.
  158:         move.l  end_adrs,d5             ; end address.
  159:         cmp.b   #$ff,d2
  160:         beq     wrfil2
  161:
  162:         move.l  #okbuf,prnadr           ; message address.
  163:         bsr     print
  164:         bsr     getchar
  165:         cmp.b   #'y',d0
  166:         beq     wrfil_2
  167:
  168:         cmp.b   #'Y',d0
  169:         beq     wrfil_2
  170:
  171:         bra     wrfil2                  ; jump, if not 'y'.
  172:
  173: wrfil_2:
  174:         cmp.l   d4,d5                   ; check size.
  175:         beq     wrfil2                  ; jump, if size = 0.
  176:
  177:         move.l  d4,adr180               ; start address.
  178:         sub.l   d4,d5
  179:         and.l   #$ffff,d5
  180:         bne     wrfil_1
  181:
  182:         move.l  #$10000,d5              ; d5.l: file size.
  183: wrfil_1:
  184:         move.l  #name1,a0               ; name pointer.
  185:         move.w  #$20,d0                 ; standard file attribute.
  186:         bsr     creat
  187:         tst.l   d0
  188:         bmi     wrfil4                  ; jump, if error.
  189:
  190:         move.l  adr180,a0               ; data pointer.
  191:         move.w  d0,fileno1              ; file handle number.
  192:
  193: wrfil_0:
  194:         move.w  fileno1,d0              ; file handle number.
  195:         move.l  d5,d1                   ; write file size.
  196:         bsr     write
  197:         tst.l   d0
  198:         bmi     wrfil3
  199:
  200:         beq     wrfil1                  ; jump, if file end.
  201:
  202:         sub.l   d0,d5                   ; write file size.
  203:         move.l  adr180,a0
  204:         add.l   d0,a0
  205:         bra     wrfil_0                 ; jump, if not file end.
  206:
  207: wrfil1:
  208:         move.w  fileno1,d0              ; write file handle number.
  209:         bsr     close
  210:         tst.l   d0
  211:         bmi     wrfil5
  212:
  213: wrfil2:
  214:         bsr     crlf
  215:         rts
  216:
  217: wrfil3:
  218:         move.l  #wemsg,prnadr
  219:         bsr     print
  220:         jmp     exit                    ; return to OS.
  221:
  222: wrfil4:
  223:         move.l  #crmsg,prnadr
  224:         bsr     print
  225:         jmp     exit                    ; return to OS.
  226:
  227: wrfil5:
  228:         move.l  #clmsg,prnadr
  229:         bsr     print
  230:         jmp     exit                    ; return to OS.
  231:
  232:
  233: **********************
  234: *  get file name     *
  235: **********************
  236: fname:
  237:         move.l  a0,-(a7)
  238:         bsr     print
  239:         move.l  (a7)+,a0
  240:
  241:         bsr     gets
  242:         move.l  #keybuf+2,a1
  243: fname0:
  244:         move.b  (a1),(a0)+
  245:         tst.b   (a1)+
  246:         bne     fname0
  247:         rts
  248:
  249:
  250: **********************
  251: *    ASCII TO BIN    *
```

▶「オムライスが食べたい」は，本当にOh!Xのスタッフの個性を象徴的に表していた感じです。それにあのイラストには笑わせてもらいました。　嵯峨　進　(19)　秋田県

```
252: ********************
253: atob:
254:         move.b  (a0)+,d0
255:         bsr     atobsub
256:         move.b  d0,d1
257:         asl.b   #4,d1
258:         move.b  (a0)+,d0
259:         bsr     atobsub
260:         or.b    d1,d0
261:         rts
262:
263:
264: atobsub:
265:         sub.b   #$30,d0
266:         cmp.b   #10,d0
267:         bcs     atobsub0
268:
269:         sub.b   #7,d0
270: atobsub0:
271:         and.b   #$0f,d0
272:         rts
273:
274:
275: ********************
276: *    CRLF, SPACE     *
277: ********************
278: crlf:
279:         move.w  d0,-(a7)
280:         move.b  #$0d,d0
281:         bsr     putchar
282:         move.b  #$0a,d0
283:         bsr     putchar
284:         move.w  (a7)+,d0
285:         rts
286:
287: space:
288:         move.w  d0,-(a7)
289:         move.b  #$20,d0
290:         bsr     putchar
291:         move.w  (a7)+,d0
292:         rts
293:
294:
295: ****************************************
296:
297: *
298: *
299: * bios function code definition
300: *
301: _exit           equ     $ff00
302: _getchar        equ     $ff01
303: _putchar        equ     $ff02
304: _print          equ     $ff09
305: _gets           equ     $ff0a
306:
307: _super          equ     $ff20
308: _creat          equ     $ff3c
309: _open           equ     $ff3d
310: _close          equ     $ff3e
311: _read           equ     $ff3f
312:
313: _write          equ     $ff40
314: _delete         equ     $ff41
315:
316:
317: ****************************************************
318:
319: *
320: * bios calls
321: *
322:
323: exit:
324:         dc.w    _exit                   ; return to os
325:
326: getchar:
327:         dc.w    _getchar                ; key in, type
328:         cmp.b   #$41,d0
329:         bcs     getchar_0
330:
331:         cmp.b   #$5b,d0
332:         bcc     getchar_0
333:
334:         or.b    #$20,d0
335: getchar_0:
336:         rts
337:
338: putchar:
339:         move.w  d0,-(sp)                ; type
340:         dc.w    _putchar
341:         addq.l  #2,sp
342:         rts
343:
344: print:
345:         move.l  (prnadr),-(sp)          ; print string
346:         dc.w    _print
347:         addq.l  #4,sp
348:         rts
349:
350: gets:
351:         pea     keybuf                  ; get string
352:         dc.w    _gets
353:         addq.l  #4,sp
354:         rts
355:
356: creat:
357:         move.w  d0,-(a7)                ; d0...atribute
358:         move.l  a0,-(a7)                ; a0...nameptr
359:         dc.w    _creat
360:         addq.l  #6,a7
361:         rts
362:
363: open:
364:         move.w  d0,-(a7)                ; d0...mode
365:         move.l  a0,-(a7)                ; a0...nameptr
366:         dc.w    _open
367:         addq.l  #6,a7
368:         rts
369:
370: close:
371:         move.w  d0,-(a7)                ; d0...file number
372:         dc.w    _close
373:         addq.l  #2,a7
374:         rts
375:
376: read:
377:         move.l  d1,-(a7)                ; d1...byte size
378:         move.l  a0,-(a7)                ; a0...dataptr
379:         move.w  d0,-(a7)                ; d0...file number
380:         dc.w    _read
381:         lea     10(a7),a7
382:         rts
383:
384: write:
385:         move.l  d1,-(a7)                ; d1...byte size
386:         move.l  a0,-(a7)                ; a0...dataptr
387:         move.w  d0,-(a7)                ; d0...file number
388:         dc.w    _write
389:         lea     10(a7),a7
390:         rts
391:
392: delete:
393:         move.l  a0,-(a7)                ; a0...nameptr
394:         dc.w    _delete
395:         addq.l  #4,a7
396:         rts
397:
398: ****************************************************
399:
400: *                               ; storage for byte data
401:
402: ****************************************************
403:
404: keybuf:
405:         dc.b    80                      ; max. input characters.
406:         dc.b    5                       ; input character number.
407:         ds.b    81                      ; data area.
408:
409:
410: ********************************************************
411:
412:         even                            ; storage for word, long data
413:
414: ********************************************************
415:
416:
417: prnadr:
418:         ds.l    1                       ; print buf. address.
419:
420:
421:
422: ****************************************************
423:
424: *                               ; storage for byte data
425:
426: ****************************************************
427:
428: *
429: okbuf:
430:         dc.b    $0d,$0a
431:         dc.b    '  ok? (y/n) ',0
432: *
433: remsg:
434:         dc.b    $0d,$0a
435:         dc.b    ' read error'
436:         dc.b    $0d,$0a,0
437:
438: *
439: wemsg:
440:         dc.b    $0d,$0a
441:         dc.b    ' write error'
442:         dc.b    $0d,$0a,0
443:
444: *
445: nfmsg:
446:         dc.b    $0d,$0a
447:         dc.b    ' not found'
448:         dc.b    $0d,$0a,0
449:
450: *
451: crmsg:
452:         dc.b    $0d,$0a
453:         dc.b    ' creat error'
454:         dc.b    $0d,$0a,0
455:
456: *
457: clmsg:
458:         dc.b    $0d,$0a
459:         dc.b    ' close error'
460:         dc.b    $0d,$0a,0
461:
462: *
463: hfnmsg:
464:         dc.b    $0d,$0a
465:         dc.b    ' input  file name ( HEX file only ) : '
466:         dc.b    0
467:
468: *
469: bfnmsg:
470:         dc.b    ' output file name ( Binary file   ) : '
471:         dc.b    0
472:
473: *
474: hexmsg:
475:         dc.b    $0d,$0a
476:         dc.b    ' HEX format error'
477:         dc.b    $0d,$0a
478:         dc.b    $0d,$0a,0
479:
480: *
481: name0:
482:         ds.b    40
483:
484: *
485: name1:
486:         ds.b    40
487:
488: *
489: count:
490:         ds.b    1
491:
492: *
493: chksum:
494:         ds.b    1
495:
496:
497: ********************************************************
498:
499:         even                    ; storage for word, long data
500:
501: ********************************************************
502:
503: *
504: fileno0:
505:         ds.w    1
506:
507: *
508: fileno1:
509:         ds.w    1
510:
511: *
512: adr180:
513:         ds.l    1
514:
515: *
516: end_adrs:
517:         ds.l    1
518:
519:
520:         end
```

リスト3 RS.S

```
================== RS.S ==================
  1: *************************
  2: *                       *
  3: *  program name : RS.S  *
  4: *                       *
  5: *   RS-232C receiver    *
  6: *                       *
  7: *   1988 9/16  by TK-YSD *
  8: *                       *
  9: *************************
 10:
 11: buff_adrs       equ     $e40000         ; TEXT VRAM address.
 12:
 13:         .text
 14:
 15:
```

▶「ROGUEスゴロク」はとても面白く，また懐かしかった。バックナンバーを引っ張り出しては，「あのころあんなことやってたなー」などと楽しみました。
戸嶋　秀和（18）大阪府

```
 16: **********************
 17: *       main         *
 18: **********************
 19: main:
 20:         clr.l   -(sp)
 21:         dc.w    _super                  ; system mode.
 22:         move.l  d0,sp
 23:
 24:         move.l  #recmsg,prnadr          ; message address.
 25:         bsr     print
 26:         bsr     rec                     ; receive rs-232c file.
 27:         bsr     wrfile                  ; save file.
 28:
 29:         move.l  #buff_adrs,a0           ; clear memory.
 30: main0:
 31:         move.l  #0,(a0)+
 32:         cmp.l   #buff_adrs+$40000,a0
 33:         bne     main0
 34:
 35:         bsr     crlf
 36:         jmp     exit                    ; return to OS.
 37:
 38:
 39: **********************
 40: *      receive       *
 41: **********************
 42: rec:
 43:         move.l  #buff_adrs,a0
 44: rec0:
 45:         bsr     reader
 46:         move.b  d0,(a0)+
 47:         cmp.b   #$1a,d0                 ; file end?
 48:         bne     rec0                    ; jump, if not end.
 49:
 50:         rts
 51:
 52:
 53: **********************
 54: *    get address     *
 55: **********************
 56: getadrs:
 57:         move.l  #buff_adrs,a0
 58: getadrs0:
 59:         tst.b   (a0)+
 60:         beq     getadrs0
 61:
 62:         move.l  a0,d4
 63:         subq.l  #1,d4                   ; start address.
 64:
 65: getadrs1:
 66:         cmp.b   #$1a,(a0)+
 67:         bne     getadrs1
 68:
 69:         move.l  a0,d5
 70:         subq.l  #1,d5                   ; end address.
 71:         rts
 72:
 73:
 74: **********************
 75: *      wrfile        *
 76: **********************
 77: wrfile:
 78:         move.l  #name1,a0               ; name pointer.
 79:         bsr     fname                   ; get file name.
 80:
 81:         bsr     getadrs                 ; get d4: start, d5: end address

 82:         cmp.b   #$ff,d2
 83:         beq     wrfil2
 84:
 85:         move.l  #okmsg,prnadr           ; message address.
 86:         bsr     print
 87:         bsr     getchar
 88:         cmp.b   #'y',d0
 89:         bne     wrfil2                  ; jump, if not 'y'.
 90:
 91:         cmp.l   d4,d5                   ; check size.
 92:         beq     wrfil2                  ; jump, if size = 0.
 93:
 94:         move.l  d4,adr180               ; start address.
 95:         addq.l  #1,d5
 96:         sub.l   d4,d5
 97:         and.l   #$ffff,d5
 98:         bne     wrfil_1
 99:
100:         move.l  #$10000,d5              ; d5.l: file size.
101: wrfil_1:
102:         move.l  #name1,a0               ; name pointer.
103:         move.w  #$20,d0                 ; standard file attribute.
104:         bsr     creat
105:         tst.l   d0
106:         bmi     wrfil4                  ; jump, if error.
107:
108:         move.l  adr180,a0               ; data pointer.
109:         move.w  d0,fileno1              ; file handle number.
110:
111: wrfil_0:
112:         move.w  fileno1,d0              ; file handle number.
113:         move.l  d5,d1                   ; write file size.
114:         bsr     write
115:         tst.l   d0
116:         bmi     wrfil3
117:
118:         beq     wrfil1                  ; jump, if file end.
119:
120:         sub.l   d0,d5                   ; write file size.
121:         move.l  adr180,a0
122:         add.l   d0,a0
123:         bra     wrfil_0                 ; jump, if not file end.
124:
125: wrfil1:
126:         move.w  fileno1,d0              ; write file handle number.
127:         bsr     close
128:         tst.l   d0
129:         bmi     wrfil5
130:
131: wrfil2:
132:         bsr     crlf
133:         rts
134:
135: wrfil3:
136:         move.l  #wemsg,prnadr
137:         bsr     print
138:         bra     wrfil1
139:
140: wrfil4:
141:         move.l  #crmsg,prnadr
142:         bsr     print
143:         rts
144:
145: wrfil5:
146:         move.l  #clmsg,prnadr
147:         bsr     print
148:         rts
149:
150:
151: **********************
152: *   get file name    *
153: **********************
154: fname:
155:         move.l  a0,-(a7)
156:         move.l  #fnmsg,prnadr           ; filename request.
157:         bsr     print
158:         move.l  (a7)+,a0
159:
160:         bsr     gets
161:         move.l  #keybuf+2,a1
162: fname0:
163:         move.b  (a1),(a0)+
164:         tst.b   (a1)+
165:         bne     fname0
166:
167:         rts
168:
169:
170: **********************
171: *       reader       *
172: **********************
173: reader:
174:         moveq.l #$32,d0                 ; inp232c.
175:         trap    #15
176: *                                       ; d0.b : data.
177:         rts
178:
179:
180: **********************
181: *    CRLF, SPACE     *
182: **********************
183: crlf:
184:         move.w  d0,-(a7)
185:         move.b  #$0d,d0
186:         bsr     putchar
187:         move.b  #$0a,d0
188:         bsr     putchar
189:         move.w  (a7)+,d0
190:         rts
191:
192: space:
193:         move.w  d0,-(a7)
194:         move.b  #$20,d0
195:         bsr     putchar
196:         move.w  (a7)+,d0
197:         rts
198:
199:
200: *********************************************
201:
202: *
203: *
204: * bios function code definition
205: *
206: _exit           equ     $ff00
207: _getchar        equ     $ff01
208: _putchar        equ     $ff02
209: _gets           equ     $ff0a
210: _print          equ     $ff09
211: _super          equ     $ff20
212: _creat          equ     $ff3c
213: _close          equ     $ff3e
214: _write          equ     $ff40
215:
216:
217: **********************************************************
218:
219: *
220: * bios calls
221: *
222:
223: exit:
224:         dc.w    _exit                   ; return to os
225:
226: getchar:
227:         dc.w    _getchar                ; key in, type
228:         cmp.b   #$41,d0
229:         bcs     getchar_0
230:
231:         cmp.b   #$5b,d0
232:         bcc     getchar_0
233:
234:         or.b    #$20,d0
235: getchar_0:
236:         rts
237:
238: putchar:
239:         move.w  d0,-(sp)                ; type
240:         dc.w    _putchar
241:         addq.l  #2,sp
242:         rts
243:
244: gets:
245:         pea     keybuf                  ; get string
246:         dc.w    _gets
247:         addq.l  #4,sp
248:         rts
249:
250: print:
251:         move.l  (prnadr),-(sp)          ; print string
252:         dc.w    _print
253:         addq.l  #4,sp
254:         rts
255:
256: creat:
257:         move.w  d0,-(a7)                ; d0...atribute
258:         move.l  a0,-(a7)                ; a0...nameptr
259:         dc.w    _creat
260:         addq.l  #6,a7
261:         rts
262:
263: close:
264:         move.w  d0,-(a7)                ; d0...file number
265:         dc.w    _close
266:         addq.l  #2,a7
267:         rts
268:
269: write:
270:         move.l  d1,-(a7)                ; d1...byte size
271:         move.l  a0,-(a7)                ; a0...dataptr
272:         move.w  d0,-(a7)                ; d0...file number
273:         dc.w    _write
274:         lea     10(a7),a7
275:         rts
276:
277: ********************************************************
278:
279: *                                ; storage for byte data
280:
281: ********************************************************
282:
283: keybuf:
284:         dc.b    80                      ; max. input characters.
285:         dc.b    5                       ; input character number.
286:         ds.b    81                      ; data area.
287:
288:
289: **************************************************************
290:
291:         even                            ; storage for word, long data
292:
293: **************************************************************
294:
295:
296: prnadr:
297:         ds.l    1                       ; print buf. address.
298:
299:
300:
301: ********************************************************
302:
303: *                                ; storage for byte data
304:
305: ********************************************************
306:
307: *
308: recmsg:
309:         dc.b    $0d,$0a
310:         dc.b    ' Receiving from RS-232C....'
```

▶「Z80マシン語工房」はいいですね。いまは受験生なので，じっくりと読んでいる時間がありませんが，受験が終わったら早くマシン語が使えるようになりたいものです。

和田　勝　(15)　長野県

```
311:         dc.b    $0d,$0a,0
312:
313: *
314: fnmsg:
315:         dc.b    $0d,$0a
316:         dc.b    ' file name: '
317:         dc.b    0
318:
319: *
320: okmsg:
321:         dc.b    $0d,$0a
322:         dc.b    '  ok? (y/n) ',0
323:
324: *
325: wemsg:
326:         dc.b    $0d,$0a
327:         dc.b    ' write error'
328:         dc.b    $0d,$0a,0
329:
330: *
331: crmsg:
332:         dc.b    $0d,$0a
333:         dc.b    ' creat error'
334:         dc.b    $0d,$0a,0
335:
336: *
337: clmsg:
338:         dc.b    $0d,$0a
339:         dc.b    ' close error'
340:         dc.b    $0d,$0a,0
341:
342: *
343: name1:
344:         ds.b    40
345:
346:
347: ****************************************************************
348:
349:         even                    ; storage for word, long data
350:
351: ****************************************************************
352:
353: *
354: fileno1:
355:         ds.w    1
356:
357: *
358: adr180:
359:         ds.l    1
360:
361:         end
```

リスト4 BIOS.MAC

```
==================  BIOS.MAC  ==================
    1:
    2: ; ********************************
    3: ; *                              *
    4: ; * program name : BIOS.MAC      *
    5: ; *                              *
    6: ; * 64180 system bios routine    *
    7: ; *                              *
    8: ; *  1988  5/04 by TK-YSD        *
    9: ; *        5/10  ver 1.1         *
   10: ; *        5/10  ver 1.2         *
   11: ; *        5/24  ver 1.3         *
   12: ; *        6/09  ver 1.4         *
   13: ; *        6/12  ver 1.5         *
   14: ; *        8/14  ver 1.6         *
   15: ; *        9/29  ver 1.7         *
   16: ; *                              *
   17: ; ********************************
   18:
   19:         .Z80
   20:         ASEG
   21:
   22:         ORG     100H
   23:
   24:
   25: PBEGIN:
   26:         .PHASE  0EA00H
   27:
   28: HSTART  EQU     0000H
   29: IOBYTE  EQU     0003H
   30: CPMDRV  EQU     0004H
   31: SYSCAL  EQU     0005H
   32: NMIADR  EQU     0066H
   33: CCP     EQU     0D400H
   34: BDOSJP  EQU     0DC06H
   35:
   36:
   37: ; ********************************
   38: ; *                              *
   39: ; *  main program starts here    *
   40: ; *                              *
   41: ; ********************************
   42:
   43: BIOSST:
   44:         JP      BOOT
   45: WBOOTE:
   46:         JP      WBOOT
   47:         JP      CONST
   48:         JP      CONIN
   49:         JP      CONOUT
   50:         JP      LIST
   51:         JP      PUNCH
   52:         JP      READER
   53:         JP      HOME
   54:         JP      SELDSK
   55:         JP      SETTRK
   56:         JP      SETSEC
   57:         JP      SETDMA
   58:         JP      READ
   59:         JP      WRITE
   60:         JP      LISTST
   61:         JP      SECTRA
   62:
   63:         DEFS    13                      ; dummy data.
   64:
   65:
   66: ; **********************
   67: ; *  work for X68000   *
   68: ; *  address=0EA40H    *
   69: ; **********************
   70:
   71: FUNCNO:
   72:         DEFS    1                       ; function number.
   73:
   74: PARAM0:
   75:         DEFS    2                       ; output parameter.
   76:
   77: PARAM1:
   78:         DEFS    2                       ; input parameter.
   79:
   80: PARDRV:
   81:         DEFS    1                       ; drive.
   82:
   83: PARTRK:
   84:         DEFS    2                       ; track.
   85:
   86: PARSEC:
   87:         DEFS    1                       ; sector.
   88:
   89: PARADR:
   90:         DEFS    2                       ; address.
   91:
   92:         DEFS    5                       ; dummy data.
   93:
   94:
   95: ; **********************
   96: ; *     boot:          *
   97: ; **********************
   98: BOOT:
   99:         DI
  100:         LD      SP,STACK
  101:         CALL    INIT
  102:
  103:         CALL    IOBYIN
  104:         XOR     A
  105:         LD      (BOTDRV),A
  106:         LD      (CPMDRV),A              ; CP/M drive no.
  107:
  108:         LD      A,(PARAM1)              ; title enable?
  109:         AND     A
  110:         JR      Z,BOOT0                 ; jump, if no.
  111:
  112:         LD      DE,TTLMES               ; CP/M title.
  113:         CALL    DEPRT
  114:
  115: BOOT0:
  116:         JR      WBOOT0
  117:
  118: INIT:
  119:         LD      B,0
  120:
  121:         LD      C,32H
  122:         IN      A,(C)
  123:         AND     0FH                     ; no memory wait, 1 i/o wait.
  124:         OUT     (C),A
  125:
  126:         LD      A,0                     ; no refresh cycle.
  127:         LD      C,36H
  128:         OUT     (C),A
  129:         RET
  130:
  131: IOBYIN:
  132:         LD      A,81H
  133:         LD      (INITIO),A
  134:         LD      (IOBYTE),A
  135:         RET
  136:
  137:
  138: ; **********************
  139: ; *     wboot:         *
  140: ; **********************
  141: ERROR:
  142:         LD      DE,WERRMS
  143:         CALL    DEPRT
  144:         CALL    CONIN
  145:
  146: ;
  147: ; bdos+ccp load
  148: ;
  149: WBOOT:
  150:         DI
  151:         LD      SP,STACK
  152:         CALL    INIT
  153:
  154:         CALL    WBOOTS
  155:         OR      A
  156:         JR      NZ,ERROR
  157:
  158: WBOOT0:
  159:         LD      HL,DBLBUF               ; load buffer, 256 bytes.
  160:         LD      (BUFADR),HL
  161:
  162:         LD      A,(INITIO)              ; set i/o byte.
  163:         LD      (IOBYTE),A
  164:
  165:         LD      A,0C3H
  166:         LD      (HSTART),A
  167:         LD      HL,WBOOTE               ; wboot ent. adrs.
  168:         LD      (HSTART+1),HL
  169:
  170:         LD      (SYSCAL),A
  171:         LD      HL,BDOSJP               ; bdos ent. adr.
  172:         LD      (SYSCAL+1),HL
  173:
  174:         LD      A,(CPMDRV)              ; login disk no.
  175:         LD      C,A
  176:         JP      CCP                     ; ccp start adr.
  177:
  178: WBOOTS:
  179: ;       CALL    CHKSAV
  180:         CALL    CHKSAV
  181:         OR      A
  182:         RET     NZ
  183:
  184:         CALL    WBSUB
  185:         RET     NZ
  186:
  187:         LD      A,(CCP)
  188:         CP      0C3H
  189:         JR      NZ,CCPERR
  190:
  191:         LD      A,(BDOSJP)
  192:         CP      0C3H
  193:         JR      NZ,CCPERR
  194:
  195:         XOR     A
  196:         RET
  197:
  198: CCPERR:
  199:         LD      A,0FFH
  200:         RET
  201:
  202: ;
  203: ; wboot sub.
  204: ;
  205: WBSUB:
  206:         LD      A,1
  207:         LD      (FUNCNO),A
  208:         CALL    NMIINT
  209:         LD      A,(PARAM1)
  210:         AND     A
  211:         RET
  212:
  213:
  214: ; **********************
  215: ; *     const:         *
  216: ; **********************
  217: CONST:
  218:         LD      HL,CONSBF
  219: CONJMP:
  220:         LD      A,(IOBYTE)
  221: IOADJP:
  222:         AND     3
  223:         ADD     A,A
  224:         LD      E,A
  225:         LD      D,0
  226:         ADD     HL,DE
  227:         LD      E,(HL)
  228:         INC     HL
  229:         LD      D,(HL)
  230:         EX      DE,HL
  231:         JP      (HL)
```


▶エレショウ'88のレポートで，文章中にある「純白の布を巻き付けただけ」のコンパニオンの写真が載ってないのは，大きな手落ちであり，今後このようなミスが2度と繰り返されないことを願う。
晴山　佳彦　(18)　千葉県

```
232:
233: CONS01:
234:            LD      A,2
235:            LD      (FUNCNO),A
236:            CALL    NMIINT
237:            LD      A,(PARAM1)
238:            AND     A
239:            RET
240:
241: CONS00:
242: LSTS00:
243: LSTS01:
244:            LD      A,0FFH
245:            RET
246:
247:
248: ; **********************
249: ; *       conin:       *
250: ; **********************
251: CONIN:
252:            LD      HL,CONIBF
253:            JP      CONJMP
254:
255: CONI01:
256:            CALL    CONS01
257:            JR      Z,CONI01
258:
259:            LD      A,3
260:            LD      (FUNCNO),A
261:            CALL    NMIINT
262:            LD      A,(PARAM1)
263:            AND     A
264:            RET
265:
266:
267: ; **********************
268: ; *       conout:      *
269: ; **********************
270: CONOUT:
271:            LD      HL,CONOBF
272:            JP      CONJMP
273:
274: LIST01:
275: CONO01:
276:            LD      A,4
277:            LD      (FUNCNO),A
278:            LD      A,C
279:            LD      (PARAM0),A
280:            CALL    NMIINT
281:            RET
282:
283:
284: ; **********************
285: ; *       list:        *
286: ; **********************
287: LIST:
288:            LD      A,(IOBYTE)
289:            RLCA
290:            RLCA
291:            LD      HL,LISTBF
292:            JP      IOADJP
293:
294: LIST10:
295:            LD      A,5
296:            LD      (FUNCNO),A
297:            LD      A,C
298:            LD      (PARAM0),A
299:            CALL    NMIINT
300:            RET
301:
302:
303: ; **********************
304: ; *      punch:        *
305: ; **********************
306: PUNCH:
307:            LD      A,(IOBYTE)
308:            RRCA
309:            RRCA
310:            RRCA
311:            RRCA
312:            LD      HL,PUNBF
313:            JP      IOADJP
314:
315: PUN00:
316: CONO00:
317: LIST00:
318:            LD      A,6
319:            LD      (FUNCNO),A
320:            LD      A,C
321:            LD      (PARAM0),A
322:            CALL    NMIINT
323:            RET
324:
325:
326: ; **********************
327: ; *       reader:      *
328: ; **********************
329: READER:
330:            LD      A,(IOBYTE)
331:            RRCA
332:            RRCA
333:            LD      HL,RDRBF
334:            JP      IOADJP
335:
336: RDR00:
337: CONI00:
338:            LD      A,7
339:            LD      (FUNCNO),A
340:            CALL    NMIINT
341:            LD      A,(PARAM1)
342:            AND     A
343:            RET
344:
345:
346: ; **********************
347: ; *       home:        *
348: ; **********************
349: HOME:
350:            CALL    CHKSAV
351:            LD      HL,0FFFFH
352:            LD      (TRKOLD),HL
353:            LD      A,0FFH
354:            LD      (SECOLD),A
355:            RET
356:
357:
358: ; **********************
359: ; *      seldsk:       *
360: ; **********************
361: SELDSK:
362:            LD      HL,0
363:            LD      DE,DRVMAX
364:            LD      A,(DE)
365:            DEC     A
366:            CP      C
367:            JR      C,ERRSEL
368:
369:            LD      A,C
370:            AND     0FH
371:            LD      L,A
372:            OR      A
373:            LD      A,(BOTDRV)
374:            JR      Z,SELCOK
375:
376:            SUB     L
377:            JR      Z,SELCOK
378:
379:            LD      A,L
380: SELCOK:
381:            LD      (DRVNEW),A
382:            ADD     A,A
383:            ADD     A,A
384:            ADD     A,A
385:            ADD     A,A
386:            LD      L,A                         ; drvnew*16.
387:            INC     DE
388:            ADD     HL,DE
389:            RET
390:
391: ERRSEL:
392:            XOR     A
393:            LD      (CPMDRV),A
394:            RET
395:
396:
397: ; **********************
398: ; *      settrk:       *
399: ; **********************
400: SETTRK:
401:            LD      (TRKNEW),BC
402:            RET
403:
404:
405: ; **********************
406: ; *      setsec:       *
407: ; **********************
408: SETSEC:
409:            LD      A,C
410:            LD      (SECNEW),A
411:            RET
412:
413:
414: ; **********************
415: ; *      setdma:       *
416: ; **********************
417: SETDMA:
418:            LD      (DMAADR),BC
419:            RET
420:
421:
422: ; **********************
423: ; *       read:        *
424: ; **********************
425: READ:
426:            CALL    RDWRT
427:            LD      A,1
428:            RET     NC
429:
430:            LDIR
431:            XOR     A
432:            RET
433:
434:
435: ; *********************
436: ; *      write:       *
437: ; *********************
438: WRITE:
439:            CALL    RDWRT
440:            LD      A,1
441:            RET     NC
442:
443:            EX      DE,HL
444:            LDIR
445:            LD      (WRTFLG),A
446:            XOR     A
447:            RET
448:
449: RDWRT:
450:            CALL    CHKSAM
451:            JR      NC,SAMWRT
452:
453:            OR      A
454:            RET     NZ
455:
456:            CALL    READ1
457:            OR      A
458:            RET     NZ
459:
460:            JR      BUSETD
461:
462: SAMWRT:
463:            LD      A,(SECNEW)
464:            LD      (SECOLD),A
465: BUSETD:
466:            LD      HL,(BUFADR)
467:            LD      A,(SECNEW)
468:            LD      BC,0080H
469:            AND     1
470:            JR      Z,$+3
471:
472:            ADD     HL,BC
473:            LD      DE,(DMAADR)
474:            SCF
475:            RET
476:
477: CHKSAM:
478:            LD      HL,DRVOLD
479:            LD      DE,DRVNEW
480:            LD      A,(DE)
481:            CP      (HL)
482:            JR      NZ,CHKSAV
483:
484:            INC     DE
485:            INC     HL
486:            LD      A,(DE)
487:            CP      (HL)                        ; track.
488:            JR      NZ,CHKSAV
489:
490:            INC     DE
491:            INC     HL
492:            LD      A,(DE)
493:            CP      (HL)                        ; track.
494:            JR      NZ,CHKSAV
495:
496:            INC     DE
497:            INC     HL
498:            LD      A,(DE)
499:            XOR     (HL)                        ; sector.
500:            AND     0FEH
501:            RET     Z
502:
503: CHKSAV:
504:            LD      A,(WRTFLG)
505:            OR      A
506:            CALL    NZ,WRITE1
507:            SCF
508:            RET
509:
510: WRITE1:
511:            LD      HL,DSKWRT
512: READWR:
513:            XOR     A
514:            LD      (WRTFLG),A
515:            JP      (HL)
516:
517: READ1:
518:            LD      HL,DRVNEW
519:            LD      DE,DRVOLD
520:            LD      BC,4
521:            LDIR
522:
523:            LD      HL,DSKRED
524:            JR      READWR
525:
526: DSKRED:
527:            CALL    PARSET
```

▶なにもSTUDIO Xに無理して投稿しなくても，『マイコンBASIC Magazinle』に投稿して採用されれば，２カ月後にはFILES Oh!Xに名前が載るんだ，と思ったりする。
郷司　謙一　(19)　神奈川県

```
528:            LD      A,13
529:            LD      (FUNCNO),A
530:            CALL    NMIINT
531:            LD      A,(PARAM1)
532:            AND     A
533:            RET
534:
535: DSKWRT:
536:            CALL    PARSET
537:            LD      A,14
538:            LD      (FUNCNO),A
539:            CALL    NMIINT
540:            LD      A,(PARAM1)
541:            AND     A
542:            RET
543:
544: PARSET:
545:            LD      A,(DRVOLD)
546:            LD      (PARDRV),A
547:            LD      HL,(TRKOLD)
548:            LD      (PARTRK),HL
549:            LD      A,(SECOLD)
550:            AND     A
551:            RRA                                     ; 1/2.
552:            LD      (PARSEC),A
553:            LD      HL,(BUFADR)
554:            LD      (PARADR),HL
555:            RET
556:
557:
558: ; **********************
559: ; *       listst:      *
560: ; **********************
561: LISTST:
562:            LD      A,(IOBYTE)
563:            RLCA
564:            RLCA
565:            LD      HL,LSTSBF
566:            JP      IOADJP
567:
568: LSTS10:
569:            LD      A,15
570:            LD      (FUNCNO),A
571:            CALL    NMIINT
572:            LD      A,(PARAM1)
573:            AND     A
574:            RET
575:
576:
577: ; **********************
578: ; *       sectra:      *
579: ; **********************
580: SECTRA:
581:            LD      H,B
582:            LD      L,C
583:            RET
584:
585:
586: ; **********************
587: ; *    subroutines     *
588: ; **********************
589: DEPRT:
590:            LD      A,(DE)
591:            OR      A
592:            RET     Z
593:
594:            PUSH    DE
595:            LD      C,A
596:            CALL    CONOUT
597:            POP     DE
598:            INC     DE
599:            JR      DEPRT
600:
601: NMIINT:
602:            PUSH    HL
603:            LD      HL,(NMIADR)
604:            LD      (INTBUF),HL
605:            LD      HL,45EDH                        ; RETN code.
606:            LD      (NMIADR),HL
607:
608:            HALT
609:
610:            NOP
611:            NOP
612:            NOP
613:            LD      HL,(INTBUF)
614:            LD      (NMIADR),HL
615:            POP     HL
616:            RET
617:
618:
619: ; **********************
620: ; *     work area      *
621: ; **********************
622: DRVOLD:
623:            DEFB    0
624: TRKOLD:
625:            DEFW    0
626: SECOLD:
627:            DEFB    0
628:
629: DRVNEW:
630:            DEFB    0
631: TRKNEW:
632:            DEFW    0
633: SECNEW:
634:            DEFB    0
635:
636: DMAADR:
637:            DEFW    0080H
638: WRTFLG:
639:            DEFB    0
640:
641: DRVMAX:
642:            DEFB    1
643:
644: ; DPH table
645: ; a>disk0       drive0
646:            DEFW    0
647:            DEFW    0
648:            DEFW    0
649:            DEFW    0
650:            DEFW    DIRBUF
651:            DEFW    DPBLK
652:            DEFW    CHK00
653:            DEFW    ALL00
654:
655:
656: ; DPBLK table
   657: DPBLK:
   658:            DEFW    32                      ; sec per track...8 kbytes per t
rack.
   659:            DEFB    4                       ; block shift.....2 kbytes per d
ata block.
   660:            DEFB    15                      ; block mask......0 to 15 sec. f
or data block.
   661:            DEFB    1                       ; extnt mask.
   662:            DEFW    255                     ; disk size-1.....256 data block

   663:            DEFW    63                      ; directory max...2 kbytes for d
irectory.
   664:            DEFB    80H                     ; alloc0.
   665:            DEFB    0                       ; alloc1.
   666:            DEFW    0                       ; check size.
   667:            DEFW    0                       ; offset.
   668:
   669: TTLMES:
   670:            DEFB    1AH
   671:            DEFB    '60k CP/M Version 2.2  Rev. 1.7 < 64180 on X68000 >',13,
10
   672:            DEFB    '                  copyright (c) by Digital Research',13,
10
   673:            DEFB    0
   674:
   675: WERRMS:
   676:            DEFB    13,10
   677:            DEFB    'reboot error.',13,10
   678:            DEFB    0
   679:
   680: LSTSBF:
   681:            DEFW    LSTS00
   682:            DEFW    LSTS01
   683:            DEFW    LSTS10
   684:            DEFW    LSTS11
   685:
   686: LISTBF:
   687:            DEFW    LIST00
   688:            DEFW    LIST01
   689:            DEFW    LIST10
   690:            DEFW    LIST11
   691:
   692: CONSBF:
   693:            DEFW    CONS00
   694:            DEFW    CONS01
   695:            DEFW    CONS10
   696:            DEFW    CONS11
   697:
   698: CONIBF:
   699:            DEFW    CONI00
   700:            DEFW    CONI01
   701:            DEFW    CONI10
   702:            DEFW    CONI11
   703:
   704: CONOBF:
   705:            DEFW    CONO00
   706:            DEFW    CONO01
   707:            DEFW    CONO10
   708:            DEFW    CONO11
   709:
   710: PUNBF:
   711:            DEFW    PUN00
   712:            DEFW    PUN01
   713:            DEFW    PUN10
   714:            DEFW    PUN11
   715:
   716: RDRBF:
   717:            DEFW    RDR00
   718:            DEFW    RDR01
   719:            DEFW    RDR10
   720:            DEFW    RDR11
   721:
   722: LSTS11:
   723: LIST11:
   724: PUN01:
   725: PUN10:
   726: PUN11:
   727: RDR01:
   728: RDR10:
   729: RDR11:
   730: CONS10:
   731: CONS11:
   732: CONI10:
   733: CONI11:
   734: CONO10:
   735: CONO11:
   736:            LD      A,(INITIO)
   737:            LD      (IOBYTE),A
   738:            LD      DE,ERMES
   739:            CALL    DEPRT
   740:            JP      0
   741:
   742: ERMES:
   743:            DEFB    13,10,'bad i/o byte',13,10,0
   744:
   745:
   746: ; **********************
   747: ; *    work area 2     *
   748: ; **********************
   749: INITIO:
   750:            DEFB    81H                     ; initialize iobyte data.
   751: INTBUF:
   752:            DEFS    2                       ; save buffer for NMI.
   753:
   754: BUFADR:
   755:            DEFS    2
   756: BOTDRV:
   757:            DEFS    1
   758: CHK00:
   759:            DEFS    16
   760: ALL00:
   761:            DEFS    40
   762: DIRBUF:
   763:            DEFS    128
   764: DBLBUF:
   765:            DEFS    256
   766:
   767:            DEFS    128                     ; stack area.
   768: STACK:
   769:            DEFB    0E5H                    ; end mark.
   770:
   771: ;
   772: BIOSED:
   773: ;
   774: ;
   775: ;
   776: ;
   777:            END
```

リスト5　CPM.S

```
==================  CPM.S  ==================
     1: ****************************
     2: *                          *
     3: *  program name : CPM.S    *
     4: *                          *
     5: *  CP/M-64180 interface    *
     6: *                          *
     7: *   1988 5/15   by TK-YSD  *
     8: *        6/26   ver 1.1    *
     9: *        7/10   ver 1.2    *
    10: *        8/21   ver 1.3    *
    11: *        9/18   ver 1.4    *
    12: *        9/24   ver 1.5    *
    13: *                          *
    14: ****************************
    15:
    16: base_adrs       equ     $ec0000         ; 64180 base address.
    17: io_adrs         equ     base_adrs+$fff8 ; 71055 base address.
    18:
    19: p_0             equ     0               ;       port_0......$ecfff8.
    20: p_1             equ     2               ;       port_1......$ecfffa.
    21: p_2             equ     4               ;       port_2......$ecfffc.
```

▶ユーフォリーの楽しさというのは，なんといっても「あっ，やったな。てめぇなんか妹でもなんでもねぇ。ビシビシ」。「私がなにをしたっていうの，ひどいわ。バシッ」という兄弟ゲンカに尽きると思います。　越智　亮　(16)　島根県

```
    22: p_ct            equ     6               ;       cntrl_reg...$ecfffe.
    23:
    24: ****************************************************************************
****
    25:
    26: func_no         equ     base_adrs+$ea40 ; function number , 1 byte.
    27: param_0         equ     base_adrs+$ea41 ; input  parameter, 2 bytes.
    28: param_1         equ     base_adrs+$ea43 ; output parameter, 2 bytes.
    29: para_drv        equ     base_adrs+$ea45 ; drive           , 1 byte.
    30: para_trk        equ     base_adrs+$ea46 ; track           , 2 bytes.
    31: para_sec        equ     base_adrs+$ea48 ; sector          , 1 byte.
    32: para_adrs       equ     base_adrs+$ea49 ; address         , 2 bytes.
    33:
    34: ****************************************************************************
****
    35:
    36:
    37:         .text
    38:
    39: **********************
    40: *     initialize     *
    41: **********************
    42: init:
    43:         lea     usersp,a7
    44:
    45:         lea     $10(a0),a0              ; set free prgram memory.
    46:         sub.l   a0,a1
    47:         move.l  a1,-(a7)
    48:         move.l  a0,-(a7)
    49:         dc.w    _setblock
    50:         addq.l  #8,a7
    51:
    52:         add.l   #$080000,a1
    53:         move.l  a1,-(a7)                ; get 512k space.
    54:         dc.w    _malloc
    55:         addq.l  #4,a7
    56:         tst.l   d0
    57:         bmi     memerr                  ; jump, if error.
    58:
    59:         move.l  d0,memptr               ; memory pointer.
    60:         add.l   #$10,d0
    61:         add.l   a1,d0
    62:         move.l  d0,disk_adrs            ; disk pointer.
    63:         clr.b   mem_type                ; select ram_type.
    64:         move.l  #io_adrs,a6             ; 71055 base address.
    65:
    66:         move.b  #$90,p_ct(a6)           ; mode0, port_0.....input.
    67: *                                       ;        port_1.....output.
    68: *                                       ;        port_2.....output.
    69:         bsr     set_res                 ; set  64180 RESET.
    70:         bsr     set_bus                 ; set  64180 BUSREQ.
    71:
    72:         clr.b   string_no               ; clear batch word counter.
    73:         move.b  (a2)+,d2                ; any option?
    74:         tst.b   d2
    75:         beq     init_7                  ; jump, if no option.
    76:
    77:         move.b  (a2)+,d0
    78:         cmp.b   #'-',d0
    79:         bne     init_7
    80:
    81:         move.b  (a2)+,d0                ; get option.
    82:         cmp.b   #'i',d0                 ; initialize?
    83:         beq     init_0                  ; jump, if yes.
    84:
    85:         cmp.b   #'I',d0                 ; initialize?
    86:         beq     init_0                  ; jump, if yes.
    87:
    88:         bra     init_4
    89:
    90: init_0:
    91:         move.l  disk_adrs,a0            ; disk memory.
    92:         move.l  #($200-1),d0            ; 2 kbytes.
    93:         clr.l   d1
    94:
    95: init_1:
    96:         move.l  d1,(a0)+
    97:         dbra    d0,init_1
    98:
    99:         move.l  disk_adrs,a0            ; disk memory.
   100:         move.l  #($800/32-1),d0         ; clear directory.
   101:         move.b  #$e5,d1
   102:
   103: init_2:
   104:         move.b  d1,(a0)
   105:         add.l   #32,a0
   106:         dbra    d0,init_2
   107:
   108:         move.l  disk_adrs,a0            ; disk memory.
   109:         move.l  #dir_tbl,a1             ; directory table address.
   110:         move.l  #(32*2-1),d0            ; 2 files (2 entries).
   111:
   112: init_3:
   113:         move.b  (a1)+,(a0)+             ; set directory data.
   114:         dbra    d0,init_3
   115:
   116: init_30:
   117:         bsr     load_cpm                ; load CP/M system.
   118:         bsr     load_cpm_file           ; load CP/M files.
   119:         bra     init_8
   120:
   121: init_4:
   122:         cmp.b   #'b',d0                 ; batch?
   123:         beq     init_5                  ; jump, if yes.
   124:
   125:         cmp.b   #'B',d0                 ; batch?
   126:         beq     init_5                  ; jump, if yes.
   127:
   128:         bra     init_6
   129:
   130: init_5:
   131:         subq.b  #3,d2
   132:         move.b  d2,string_no            ; number  of parameter character
s.
   133:         addq.l  #1,a2
   134:         move.l  a2,string_adrs          ; address of p.c.
   135:
   136:         bra     init_7
   137:
   138: init_6:
   139:         cmp.b   #'t',d0                 ; title enable?
   140:         beq     init_30                 ; jump, if yes.
   141:
   142:         cmp.b   #'T',d0                 ; title enable?
   143:         beq     init_30                 ; jump, if yes.
   144:
   145: init_7:
   146:         clr.b   param_1                 ; title disable.
   147:         bra     init_9
   148:
   149: init_8:
   150:         move.b  #$ff,param_1            ; title enable.
   151: init_9:
   152:         bsr     set_res                 ; set  64180 RESET.
   153:         bsr     set_opr                 ; set  64180 OPERATION.
   154:         clr.b   filemode                ; set  file close mode.
   155:
   156: **********************
   157: *       main         *
   158: **********************
   159: main:
   160:         bsr     wait_hlt                ; wait halt.
   161:
   162:         bsr     set_bus                 ; set  64180 BUSREQ.
   163:         bsr     bios                    ; bios function.
   164:         bsr     set_nmi                 ; set  64180 NMI.
   165:         bra     main
   166:
```

```
   167: escape:
   168:         bsr     set_res                 ; set  64180 RESET.
   169:         bsr     set_bus                 ; set  64180 BUSREQ.
   170:         bsr     crlf
   171:         dc.w    _exit                   ; return to OS.
   172:
   173: memerr:
   174:         move.w  #1,-(a7)
   175:         dc.w    _exit2                  ; return to OS.
   176:
   177:
   178: **********************
   179: *  bios subroutine   *
   180: **********************
   181: bios:
   182:         move.b  func_no,d0              ; function number.
   183:         move.b  #$ff,func_no
   184:
   185:         cmp.b   #0,d0
   186:         beq     escape
   187:
   188:         cmp.b   #1,d0
   189:         beq     load_ccp_bdos
   190:
   191:         cmp.b   #2,d0
   192:         beq     const
   193:
   194:         cmp.b   #3,d0
   195:         beq     conin
   196:
   197:         cmp.b   #4,d0
   198:         beq     conout
   199:
   200:         cmp.b   #5,d0
   201:         beq     list
   202:
   203:         cmp.b   #6,d0
   204:         beq     punch
   205:
   206:         cmp.b   #7,d0
   207:         beq     reader
   208:
   209:         cmp.b   #13,d0
   210:         beq     read_rec
   211:
   212:         cmp.b   #14,d0
   213:         beq     write_rec
   214:
   215:         cmp.b   #15,d0
   216:         beq     listst
   217:
   218:         cmp.b   #16,d0
   219:         beq     rdfile
   220:
   221:         cmp.b   #17,d0
   222:         beq     wrfile
   223:
   224:         move.l  #ifmsg,prnadrs
   225:         bsr     print
   226:         rts
   227:
   228:
   229: **********************
   230: *   load_ccp_bdos    *
   231: **********************
   232: load_ccp_bdos:
   233:         move.l  #(base_adrs+$d400),address
   234:         move.l  #name1,a0               ; name pointer.
   235:         bsr     load_x
   236:         clr.b   param_1
   237:         rts
   238:
   239:
   240: **********************
   241: *       const        *
   242: **********************
   243: const:
   244:         tst.b   string_no
   245:         beq     const_0
   246:
   247:         bra     const_1
   248:
   249: const_0:
   250:         move.w  #$fe,-(a7)              ; key sense.
   251:         dc.w    _inpout
   252:         addq.l  #2,a7
   253:
   254:         tst.b   d0
   255:         beq     const_2
   256:
   257: const_1:
   258:         move.b  #$ff,d0
   259: const_2:
   260:         move.b  d0,param_1
   261:         rts
   262:
   263:
   264: **********************
   265: *       conin        *
   266: **********************
   267: conin:
   268:         tst.b   string_no
   269:         beq     conin_1
   270:
   271:         subq.b  #1,string_no
   272:         move.l  a0,-(a7)
   273:         move.l  string_adrs,a0
   274:         move.b  (a0)+,d0
   275:         cmp.b   #'@',d0
   276:         bne     conin_0
   277:
   278:         move.b  #$0d,d0
   279:
   280: conin_0:
   281:         bsr     caps
   282:         move.b  d0,param_1
   283:         move.l  a0,string_adrs
   284:         move.l  (a7)+,a0
   285:         rts
   286:
   287: conin_1:
   288:         move.w  #$ff,-(a7)              ; key input.
   289:         dc.w    _inpout
   290:         addq.l  #2,a7
   291:
   292:         move.b  d0,param_1
   293:         rts
   294:
   295: caps:
   296:         cmp.b   #$61,d0
   297:         bcs     caps_0
   298:
   299:         cmp.b   #$7b,d0
   300:         bcc     caps_0
   301:
   302:         and.b   #$df,d0
   303: caps_0:
   304:         rts
   305:
   306:
   307: **********************
   308: *       conout       *
   309: **********************
   310: conout:
   311:         move.b  param_0,d0
   312:         cmp.b   #$fe,d0
   313:         bcc     conout_0
   314:
```

▶最近のテレビ番組のなかでよく，ソーサリアンのゲームミュージックが流れているので，よく聞いてみよう。　　西沢　直樹　(17)　長野県

```
315:         move.w  d0,-(a7)                ; char. output.
316:         dc.w    _inpout
317:         addq.l  #2,a7
318:
319: conout_0:
320:         rts
321:
322:
323: **********************
324: *       list         *
325: **********************
326: list:
327:         clr.w   d0
328:         move.b  param_0,d0
329:
330:         move.w  d0,-(a7)                ; print out.
331:         dc.w    _prnout
332:         addq.l  #2,a7
333:
334:         rts
335:
336:
337: **********************
338: *       punch        *
339: **********************
340: punch:
341:         moveq.l #$35,d0                 ; out232c.
342:         move.b  param_0,d1              ; data.
343:         trap    #15
344:         rts
345:
346:
347: **********************
348: *       reader       *
349: **********************
350: reader:
351:         moveq.l #$32,d0                 ; inp232c.
352:         trap    #15
353:         move.b  d0,param_1
354:         rts
355:
356:
357: **********************
358: *     read_rec       *
359: **********************
360: read_rec:
361:         move.b  #$ff,param_1            ; NG.
362:         tst.b   para_drv                ; A drive?
363:         bne     read_rec_1              ; jump, if no.
364:
365:         bsr     get_adrs
366:         move.l  #255,d0                 ; 256 bytes.
367:
368: read_rec_0:
369:         move.b  (a0)+,(a1)+
370:         dbra    d0,read_rec_0
371:
372:         clr.b   param_1                 ; OK.
373: read_rec_1:
374:         rts
375:
376: get_adrs:
377:         clr.l   d0
378: *       move.b  para_trk+1,d0
379: *       asl.l   #8,d0
380:         move.b  para_trk,d0
381:         and.b   #$7f,d0                 ; 512 kbytes max.
382:         asl.l   #4,d0
383:
384:         clr.l   d1
385:         move.b  para_sec,d1
386:         and.b   #$0f,d1
387:         add.l   d1,d0
388:         asl.l   #8,d0
389:         add.l   disk_adrs,d0
390:         move.l  d0,a0                   ; 68000 address.
391:
392:         clr.l   d0
393:         move.b  para_adrs+1,d0
394:         asl.l   #8,d0
395:         move.b  para_adrs,d0
396:         add.l   #base_adrs,d0
397:         move.l  d0,a1                   ; 64180 address.
398:         rts
399:
400:
401: **********************
402: *     write_rec      *
403: **********************
404: write_rec:
405:         move.b  #$ff,param_1            ; NG.
406:         tst.b   para_drv                ; A drive?
407:         bne     write_rec_1             ; jump, if no.
408:
409:         bsr     get_adrs
410:         move.l  #255,d0                 ; 256 bytes.
411:
412: write_rec_0:
413:         move.b  (a1)+,(a0)+
414:         dbra    d0,write_rec_0
415:
416:         clr.b   param_1                 ; OK.
417: write_rec_1:
418:         rts
419:
420:
421: **********************
422: *      listst        *
423: **********************
424: listst:
425:         dc.w    _prnsns                 ; printer status.
426:
427:         move.b  d0,param_1
428:         rts
429:
430:
431: **********************
432: *      load_cpm      *
433: **********************
434: load_cpm:
435:         move.l  #base_adrs,a0           ; clear $0000 to $0FFF7H.
436:         move.l  #($10000-8-1),d0
437:         clr.b   d1
438:
439: load_cpm_5:
440:         move.b  d1,(a0)+
441:         dbra    d0,load_cpm_5
442:
443:         bsr     load_ccp_bdos           ; load CCP, BDOS.
444:         move.l  #(base_adrs+$ea00),address
445: *                                       ; load BIOS.
446:         move.l  #name0,a0               ; name pointer.
447:
448: load_x:
449:         clr.w   d0                      ; read file mode.
450:         bsr     open
451:         tst.l   d0
452:         bmi     load_cpm_4              ; jump, if error.
453:
454:         move.w  d0,fileno0              ; file handle number.
455:
456: load_cpm_0:
457:         move.l  address,a0              ; data pointer.
458:         move.w  fileno0,d0              ; file handle number.
459:         move.l  #65528,d1               ; read file size.
460:         bsr     read
461:         tst.l   d0
462:         bmi     load_cpm_3              ; jump, if error.
463:
464:         beq     load_cpm_1              ; jump, if file end.
465:
466:         add.l   d0,address
467:         bra     load_cpm_0
468:
469: load_cpm_1:
470:         move.w  fileno0,d0              ; read file handle number.
471:         bsr     close
472:
473: jump_set:
474:         move.l  #base_adrs,a0           ; set 'JP 0EA00H' to $0000.
475:         move.b  #$c3,(a0)+
476:         move.b  #$00,(a0)+
477:         move.b  #$ea,(a0)+
478:         rts
479:
480: load_cpm_3:
481:         move.l  #remsg,prnadrs
482:         bsr     print
483:         bra     load_cpm_1
484:
485: load_cpm_4:
486:         move.l  #nfmsg,prnadrs
487:         bsr     print
488:         rts
489:
490:
491: **********************
492: *   load_cpm_file    *
493: **********************
494: load_cpm_file:
495:         clr.l   d0
496:         move.b  dir_tbl+32*0+16,d0
497:         move.l  #name_ex,a0             ; name pointer.
498:         bsr     load
499:
500:         clr.l   d0
501:         move.b  dir_tbl+32*1+16,d0
502:         move.l  #name_os,a0             ; name pointer.
503:         bsr     load
504:
505:         rts
506:
507:
508: **********************
509: *       load         *
510: **********************
511: load:
512:         asl.l   #8,d0                   ; get save address.
513:         asl.l   #3,d0
514:         add.l   disk_adrs,d0
515:         move.l  d0,address
516:         clr.w   d0                      ; read file mode.
517:         bsr     open
518:         tst.l   d0
519:         bmi     load_cpm_4              ; jump, if error.
520:
521:         move.w  d0,fileno0              ; file handle number.
522:
523: load_0:
524:         move.l  address,a0              ; data pointer.
525:         move.w  fileno0,d0              ; file handle number.
526:         move.l  #65528,d1               ; read file size.
527:         bsr     read
528:         tst.l   d0
529:         bmi     load_cpm_3              ; jump, if error.
530:
531:         beq     load_2                  ; jump, if file end.
532:
533:         add.l   d0,address
534:         bra     load_0
535:
536: load_2:
537:         move.w  fileno0,d0              ; read file handle number.
538:         bsr     close
539:         rts
540:
541:
542: **********************
543: *      rdfile        *
544: **********************
545: rdfile:
546:         tst.b   filemode
547:         bne     rdfile_1                ; jump, if already opened.
548:
549:         move.l  #name20,a0              ; get file name.
550:         move.l  #(base_adrs+$1000),a1   ; file name buf.
551:         move.l  #29,d0                  ; 30 bytes.
552:
553: rdfile_0:
554:         move.b  (a1)+,(a0)+
555:         dbra    d0,rdfile_0
556:
557:         move.l  #name2,a0               ; name pointer.
558:         clr.w   d0                      ; read file mode.
559:         bsr     open
560:         tst.l   d0
561:         bmi     rdfile_4                ; jump, if error.
562:
563:         move.w  d0,fileno0              ; file handle number.
564:         move.b  #1,filemode             ; set file open mode.
565:
566: rdfile_1:
567:         move.l  #(base_adrs+$1020),a0   ; file start address.
568:         move.w  fileno0,d0              ; file handle number.
569:         move.l  #32768,d1               ; read file size.
570:         bsr     read
571:         clr.b   param_1                 ; ok mark.
572:         tst.l   d0
573:         bmi     rdfile_2                ; jump, if error.
574:
575:         beq     rdfile_3                ; jump, if file end.
576:
577:         move.b  d0,(base_adrs+$101e)
578:         asr.w   #8,d0
579:         move.b  d0,(base_adrs+$101f)
580:         rts
581:
582: rdfile_2:
583:         move.b  #$ff,param_1            ; error mark.
584:         move.l  #remsg,prnadrs
585:         bsr     print
586:
587: rdfile_3:
588:         clr.b   filemode                ; set file close mode.
589:         clr.b   (base_adrs+$101e)
590:         clr.b   (base_adrs+$101f)
591:         move.w  fileno0,d0              ; read file handle number.
592:         bsr     close
593:         rts
594:
595: rdfile_4:
596:         move.b  #$ff,param_1            ; error mark.
597:         move.l  #nfmsg,prnadrs
598:         bsr     print
599:         rts
600:
601:
602: **********************
603: *      wrfile        *
604: **********************
605: wrfile:
606:         tst.b   filemode
607:         bne     wrfile_1                ; jump, if already opened.
608:
609:         move.l  #name20,a0              ; get file name.
610:         move.l  #(base_adrs+$1000),a1   ; file name buf.
```

▶"さよならLIVE in '88" って書いてあるから終わってしまうのかと思った。今度からは "LIVE in '89" となるんでしょ。びっくりさせないでくださいよ！

澤田　光宏　(18)　滋賀県

```
611:         move.l  #29,d0                  ; 30 bytes.
612:
613: wrfile_0:
614:         move.b  (a1)+,(a0)+
615:         dbra    d0,wrfile_0
616:
617:         move.l  #name2,a0               ; create file.
618:         move.w  #$20,d0                 ; standard file attribute.
619:         bsr     creat
620:         tst.l   d0
621:         bmi     wrfile_4                ; jump, if error.
622:
623:         move.w  d0,fileno0              ; file handle number.
624:         move.b  #2,filemode             ; set file creat mode.
625:
626: wrfile_1:
627:         clr.l   d1                      ; get file size.
628:         move.b  (base_adrs+$101f),d1
629:         asl.l   #8,d1
630:         move.b  (base_adrs+$101e),d1    ; d1: file size.
631:
632:         clr.b   param_1                 ; ok mark.
633:         cmp.w   #0,d1                   ; file end?
634:         beq     wrfile_3                ; jump, if yes.
635:
636:         move.l  #(base_adrs+$1020),a0   ; file start address.
637:         move.w  fileno0,d0              ; file handle number.
638:         bsr     write
639:         tst.l   d0
640:         bmi     wrfile_2
641:
642:         rts
643:
644:
645: wrfile_2:
646:         move.b  #$ff,param_1            ; error mark.
647:         move.l  #wemsg,prnadrs
648:         bsr     print
649:
650: wrfile_3:
651:         clr.b   filemode                ; set  file close mode.
652:         move.w  fileno0,d0              ; read file handle number.
653:         bsr     close
654:         rts
655:
656: wrfile_4:
657:         move.b  #$ff,param_1            ; error mark.
658:         move.l  #crmsg,prnadrs
659:         bsr     print
660:         rts
661:
662:
663: **********************
664: *    CRLF, SPACE     *
665: **********************
666: crlf:
667:         move.w  d0,-(a7)
668:         move.b  #$0d,d0
669:         bsr     putchar
670:         move.b  #$0a,d0
671:         bsr     putchar
672:         move.w  (a7)+,d0
673:         rts
674:
675: space:
676:         move.w  d0,-(a7)
677:         move.b  #$20,d0
678:         bsr     putchar
679:         move.w  (a7)+,d0
680:         rts
681:
682:
683: **********************
684: *  set 64180 RESET   *
685: **********************
686: set_res:
687:         move.b  #$0f,d0                 ; set 64180 RESET, ROM.
688:
689:         btst.b  #0,mem_type             ; check memory.
690:         bne     set_res_0               ; jump, if rom.
691:
692:         move.b  #$17,d0                 ; set 64180 RESET, RAM.
693: set_res_0:
694:         move.b  d0,p_1(a6)
695:         nop
696:         rts
697:
698:
699: **********************
700: *  set 64180 BUSREQ  *
701: **********************
702: set_bus:
703:         move.b  #$0c,d0                 ; set 64180 BUSREQ, ROM.
704:
705:         btst.b  #0,mem_type             ; check memory.
706:         bne     set_bus_0               ; jump, if rom.
707:
708:         move.b  #$14,d0                 ; set 64180 BUSREQ, RAM.
709: set_bus_0:
710:         move.b  d0,p_1(a6)
711:         nop
712:         bsr     wait_ack
713:         rts
714:
715:
716: **********************
717: *   set 64180 NMI    *
718: **********************
719: set_nmi:
720:         move.b  #$12,p_1(a6)            ; set   64180 NMI.
721:         nop
722:         nop
723:         nop
724:         nop
725:         nop
726:         move.b  #$16,p_1(a6)            ; clear 64180 NMI.
727:         rts
728:
729:
730: ***********************
731: * set 64180 OPERATION *
732: ***********************
733: set_opr:
734:         move.b  #$16,p_1(a6)            ; set 64180 OPERATION.
735:
736: set_opr_0:
737:         nop
738:         nop
739:         nop
740:         nop
741:         nop
742:         btst    #0,p_0(a6)              ; BUSACK on?
743:         beq     set_opr_0               ; jump, if yes.
744:
745:         nop
746:         nop
747:         nop
748:         nop
749:         nop
750:         btst    #0,p_0(a6)              ; BUSACK on?
751:         beq     set_opr_0               ; jump, if yes.
752:
753:         rts
754:
755:
756: **********************
757: * wait 64180 BUSACK  *
758: **********************
759: wait_ack:
760:         nop
761:         nop
762:         nop
763:         nop
764:         nop
765:         btst    #0,p_0(a6)              ; BUSACK on?
766:         bne     wait_ack                ; jump, if no.
767:
768:         nop
769:         nop
770:         nop
771:         nop
772:         nop
773:         btst    #0,p_0(a6)              ; BUSACK on?
774:         bne     wait_ack                ; jump, if no.
775:
776:         rts
777:
778: **********************
779: *  wait 64180 HALT   *
780: **********************
781: wait_hlt:
782:         nop
783:         nop
784:         nop
785:         nop
786:         nop
787:         btst    #1,p_0(a6)              ; HALT on?
788:         bne     wait_hlt                ; jump, if no.
789:
790:         nop
791:         nop
792:         nop
793:         nop
794:         nop
795:         btst    #1,p_0(a6)              ; HALT on?
796:         bne     wait_hlt                ; jump, if no.
797:
798:         rts
799:
800:
801: *
802: *
803: * bios function code definition
804: *
805: _exit           equ     $ff00
806: _getchar        equ     $ff01
807: _putchar        equ     $ff02
808: _cominp         equ     $ff03
809: _comout         equ     $ff04
810: _prnout         equ     $ff05
811: _inpout         equ     $ff06
812: _print          equ     $ff09
813: _prnsns         equ     $ff11
814: _super          equ     $ff20
815: _creat          equ     $ff3c
816: _open           equ     $ff3d
817: _close          equ     $ff3e
818: _read           equ     $ff3f
819: _write          equ     $ff40
820: _malloc         equ     $ff48
821: _mfree          equ     $ff49
822: _setblock       equ     $ff4a
823: _exit2          equ     $ff4c
824:
825: **********************************************************
826:
827: getchar:
828:         dc.w    _getchar                ; key in, type.
829:         rts
830:
831: putchar:
832:         move.w  d0,-(a7)                ; type.
833:         dc.w    _putchar
834:         addq.l  #2,a7
835:         rts
836:
837: print:
838:         move.l  prnadrs,-(a7)           ; print string.
839:         dc.w    _print
840:         addq.l  #4,a7
841:         rts
842:
843: creat:
844:         move.w  d0,-(a7)                ; d0...atribute.
845:         move.l  a0,-(a7)                ; a0...nameptr.
846:         dc.w    _creat
847:         addq.l  #6,a7
848:         rts
849:
850: open:
851:         move.w  d0,-(a7)                ; d0...mode.
852:         move.l  a0,-(a7)                ; a0...nameptr.
853:         dc.w    _open
854:         addq.l  #6,a7
855:         rts
856:
857: close:
858:         move.w  d0,-(a7)                ; d0...file number.
859:         dc.w    _close
860:         addq.l  #2,a7
861:         rts
862:
863: read:
864:         move.l  d1,-(a7)                ; d1...byte size.
865:         move.l  a0,-(a7)                ; a0...dataptr.
866:         move.w  d0,-(a7)                ; d0...file number.
867:         dc.w    _read
868:         lea     10(a7),a7
869:         rts
870:
871: write:
872:         move.l  d1,-(a7)                ; d1...byte size.
873:         move.l  a0,-(a7)                ; a0...dataptr.
874:         move.w  d0,-(a7)                ; d0...file number.
875:         dc.w    _write
876:         lea     10(a7),a7
877:         rts
878:
879:
880: **********************
881: *     work area      *
882: **********************
883:
884:         .data
885:
886: clmsg:
887:         dc.b    $0d,$0a
888:         dc.b    ' close error.'
889:         dc.b    $0d,$0a,0
890:
891: crmsg:
892:         dc.b    $0d,$0a
893:         dc.b    ' creat error.'
894:         dc.b    $0d,$0a,0
895:
896: ifmsg:
897:         dc.b    $0d,$0a
898:         dc.b    ' irregal func. call.'
899:         dc.b    $0d,$0a,0
900:
901: nfmsg:
902:         dc.b    $0d,$0a
903:         dc.b    ' not found.'
904:         dc.b    $0d,$0a,0
905:
906: remsg:
```

▶MusicBASICにはとても驚いた。こんなことができたらいいなぁ，と思っていた機能がほとんど付いていた。
高木　真一　(17)　岐阜県

```
907:            dc.b    $0d,$0a
908:            dc.b    ' read error.'
909:            dc.b    $0d,$0a,0
910:
911: wemsg:
912:            dc.b    $0d,$0a
913:            dc.b    ' write error.'
914:            dc.b    $0d,$0a,0
915:
916: name0:
917:            dc.b    '¥cpm¥sys¥BIOS.OBJ',0
918:
919: name1:
920:            dc.b    '¥cpm¥sys¥CCPBDOS.OBJ',0
921:
922: name2:
923:            dc.b    '¥cpm¥'
924: name20:
925:            ds.b    30
926:
927: name_ex:
928:            dc.b    '¥cpm¥EX.COM',0
929:
930: name_os:
931:            dc.b    '¥cpm¥OS.COM',0
932:
933:
934: dir_tbl:
935:            dc.b    0,'EX      COM',0,0,0,$0c
936:            dc.b    $01,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,
$00,$00
937:
938:            dc.b    0,'OS      COM',0,0,0,$02
939:            dc.b    $02,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,$00,
$00,$00
940:
941: mem_type:
942:            dc.b    0                       ; select ram_type.
943:
944: string_no:
945:            ds.b    1
946:
947: filemode:
948:            ds.b    1
949:
950:
951: **********************
952: *     work area 2    *
953: **********************
954:
955:            .even
956:
957: fileno0:
958:            ds.w    1
959:
960: memptr:
961:            ds.l    1
962:
963: disk_adrs:
964:            ds.l    1
965:
966: string_adrs:
967:            ds.l    1
968:
969: prnadrs:
970:            ds.l    1                       ; print buf. address.
971:
972: address:
973:            ds.l    1
974:
975:
976: **********************************************
977:
978:            .bss
979:
980:            ds.l    256
981:
982: usersp:
983:
984:            .end
```

リスト6 OS.MAC

```
==================  OS.MAC  ==================
     1:
     2: ; ******************************
     3: ; *                            *
     4: ; *  program name : OS.MAC     *
     5: ; *                            *
     6: ; *  escape to Human68k        *
     7: ; *                            *
     8: ; *  1988  5/29  by TK-YSD     *
     9: ; *                            *
    10: ; ******************************
    11:
    12:         .Z80
    13:         ASEG
    14:
    15:         ORG     100H
    16:
    17: BDOS    EQU     0005H           ; bdos call.
    18: FUNCNO  EQU     0EA40H          ; function number.
    19:
    20: MAIN:
    21:         LD      HL,0080H        ; any option?
    22:         LD      A,(HL)
    23:         INC     HL
    24:         AND     A
    25:         JR      Z,MAIN0         ; jump, if no.
    26:
    27:         LD      A,(HL)
    28:         INC     HL
    29:         CP      ' '
    30:         JR      NZ,MAIN0
    31:
    32:         LD      A,(HL)
    33:         INC     HL
    34:         CP      '-'
    35:         JR      NZ,MAIN0
    36:
```

```
    37:         LD      A,(HL)
    38:         INC     HL
    39:         CP      'n'
    40:         JR      Z,MAIN1
    41:
    42:         CP      'N'
    43:         JR      Z,MAIN1
    44:
    45: MAIN0:
    46:         LD      DE,DESC
    47:         CALL    MESSAG
    48:
    49: MAIN1:
    50:         LD      A,0
    51:         LD      (FUNCNO),A
    52:         HALT
    53:
    54: MESSAG:
    55:         PUSH    BC
    56:         PUSH    DE
    57:         PUSH    HL
    58:         LD      C,9
    59:         CALL    BDOS
    60:         POP     HL
    61:         POP     DE
    62:         POP     BC
    63:         RET
    64: ;
    65: DESC:
    66:         DEFB    ' good bye!'
    67:         DEFB    0DH,0AH,'$'
    68:
    69:         DEFB    0E5H                    ; end mark.
    70: ;
    71: ;
    72:         END
```

リスト7 EX.MAC

```
==================  EX.MAC  ==================
     1: ; ******************************
     2: ; *                            *
     3: ; *  program name : EX.MAC     *
     4: ; *                            *
     5: ; *  64180 <--> 68000          *
     6: ; *   file exchange program    *
     7: ; *                            *
     8: ; *  1988  5/29  by TK-YSD     *
     9: ; *        6/26  ver 1.1       *
    10: ; *        9/24  ver 1.2       *
    11: ; *                            *
    12: ; ******************************
    13:
    14:         .Z80
    15:         ASEG
    16:
    17:         ORG     100H
    18:
    19: CR      EQU     0DH
    20: LF      EQU     0AH
    21: CAN     EQU     18H
    22: EOF     EQU     1AH
    23: ESC     EQU     1BH
    24:
    25: BDOS    EQU     0005H
    26: NMIADRS EQU     0066H
    27: FNAME   EQU     1000H           ; file name.
    28: FSIZE   EQU     101EH           ; file size.
    29: BUFFER  EQU     1020H           ; file buffer address.
    30: FUNCNO  EQU     0EA40H          ; function number.
    31: PARAM1  EQU     0EA43H          ; answer from X68000.
    32:
    33:
    34: ; ******************************
    35: ; *                            *
    36: ; *  main program starts here  *
    37: ; *                            *
    38: ; ******************************
    39:
    40: MAIN:
    41:         LD      SP,STACK
    42:
    43:         CALL    OPTION          ; check command option
    44:         LD      A,(OPTFLG)
    45:         AND     A
    46:         JR      Z,MAIN1
    47:
    48:         CP      1
    49:         JR      NZ,MAIN0
    50:
    51:         CALL    LOAD
    52:         JP      0
    53:
    54: MAIN0:
    55:         CP      2
    56:         JR      NZ,MAIN1
    57:
    58:         CALL    SAVE
    59:         JP      0
    60:
    61: MAIN1:
    62:         LD      DE,DEX
    63:         CALL    MESSAG
    64:         CALL    HELP
    65: MAIN2:
    66:         LD      HL,MAIN2        ; return address.
    67:         PUSH    HL
    68:         LD      A,'*'
    69:         CALL    TYPE
    70:         CALL    KEY
    71:         CP      'H'             ; help menue.
    72:         JP      Z,HELP
    73:
    74:         CP      'R'
    75:         JP      Z,LOAD
    76:
    77:         CP      'W'
    78:         JP      Z,SAVE
    79:
    80:         CP      'E'             ; escape to CP/M.
    81:         JP      Z,0
    82:
    83:         CALL    CRLF
    84:         RET
    85:
    86: ;
    87: ; OPTION
    88: ;
    89: OPTION:
    90:         LD      A,0
    91:         LD      (OPTFLG),A      ; no option (default).
    92:
    93:         LD      BC,16           ; option file name.
    94:         LD      DE,FCB
    95:         LD      HL,006CH
    96:         LDIR
    97:
    98:         LD      HL,0080H
    99:         LD      A,(HL)          ; any option?
   100:         INC     HL
   101:         AND     A
   102:         RET     Z
   103:
   104:         LD      A,(HL)
   105:         INC     HL
   106:         CP      ' '
   107:         RET     NZ
   108:
   109:         LD      A,(HL)
   110:         INC     HL
   111:         CP      '-'
   112:         RET     NZ
   113:
```

▶TBSのザ・ベストテンが3月で終わってしまうらしい。僕はとても悲しい。
山崎　哲也　(17)　長野県

```
114:         LD      A,(HL)          ; read file?
115:         INC     HL
116:         CALL    CAPS
117:
118:         CP      'R'
119:         JR      Z,OPT0
120:
121:         CP      'W'
122:         JR      Z,OPT1
123:
124:         RET
125:
126: OPT0:
127:         LD      A,1             ; read flag.
128:         JR      OPT2
129: OPT1:
130:         LD      A,2             ; write flag.
131: OPT2:
132:         LD      (OPTFLG),A
133:
134:         INC     HL
135:         LD      DE,FNAME
136: OPT3:
137:         LD      A,(HL)
138:         INC     HL
139:         CALL    CAPS
140:         LD      (DE),A
141:         INC     DE
142:         AND     A
143:         RET     Z
144:
145:         JR      OPT3
146:
147: ; HELP MENUE
148:
149: HELP:
150:         CALL    TYPE
151:         CALL    CRLF
152:         LD      DE,DCOMND
153:         CALL    MESSAG
154:         RET
155:
156: ; FILE INPUT
157:
158: LOAD:
159:         LD      A,(OPTFLG)
160:         AND     A
161:         JR      NZ,LOAD0        ; jump, if option.
162:
163:         CALL    CRLF
164:         LD      DE,DREAD
165:         CALL    MESSAG
166:
167:         CALL    FILNAM          ; get file name.
168:         LD      DE,DREADY
169:         CALL    MESSAG
170:         CALL    KEY
171:         CALL    CRLF
172:         CP      'Y'
173:         JR      NZ,LOAD4
174:
175: LOAD0:
176:         CALL    DELETE
177:         CALL    MAKE
178:         INC     A
179:         JP      Z,DISKER
180:
181: LOAD1:
182:         CALL    FILINP          ; recieve file on memory.
183:         CALL    CALSIZ
184:         LD      A,B
185:         OR      C
186:         JR      Z,LOAD3         ; jump, if file end.
187:
188:         LD      HL,BUFFER
189: LOAD2:
190:         PUSH    BC
191:         LD      BC,80H
192:         LD      DE,80H
193:         LDIR
194:         POP     BC
195:
196:         CALL    WRITSQ
197:         OR      A
198:         JP      NZ,DISKER
199:
200:         DEC     BC
201:         LD      A,B
202:         OR      C
203:         JR      NZ,LOAD2
204:
205:         JR      LOAD1
206:
207: LOAD3:
208:         CALL    CLOSE
209:
210: LOAD4:
211:         LD      A,(OPTFLG)
212:         AND     A
213:         RET     NZ
214:
215:         CALL    CRLF
216:         RET
217:
218: ;
219: ; FILE OUTPUT
220: ;
221: SAVE:
222:         LD      A,(OPTFLG)
223:         AND     A
224:         JR      NZ,SAVE0        ; jump, if option.
225:
226:         CALL    CRLF
227:         LD      DE,DSAVE
228:         CALL    MESSAG
229:
230:         CALL    FILNAM          ; get file name.
231:         LD      DE,DREADY
232:         CALL    MESSAG
233:         CALL    KEY
234:         CALL    CRLF
235:         CP      'Y'
236:         JR      NZ,SAVE3
237:
238: SAVE0:
239:         CALL    OPEN
240:         INC     A
241:         JP      Z,OPNERR        ; jump, if no file.
242:
243:         LD      DE,BUFFER
244: SAVE1:
245:         CALL    READSQ
246:         OR      A
247:         JR      NZ,SAVE2        ; jump, if file end.
248:
249:         LD      HL,80H
250:         LD      BC,80H
251:         LDIR
252:
253:         LD      A,D
254:         CP      HIGH (BUFFER+8000H)
255:         JR      NZ,SAVE1        ; jump, if size is not 32 kbytes.
256:
257:         CALL    FILOUT          ; transmit file on memory.
258:         LD      DE,BUFFER
259:         JR      SAVE1
260:
261: SAVE2:
262:         LD      HL,BUFFER
263:         AND     A
264:         SBC     HL,DE
265:         JR      Z,SAVE20        ; jump, if size 0.
266:
267:         CALL    FILOUT          ; transmit file on memory.
268: SAVE20:
269:         LD      DE,BUFFER       ; file end, FSIZE : 0.
270:         CALL    FILOUT
271: SAVE3:
272:         LD      A,(OPTFLG)
273:         AND     A
274:         RET     NZ
275:
276:         CALL    CRLF
277:         RET
278:
279: ;
280: ; CALCULATE FILE SIZE
281: ;
282: CALSIZ:
283:         LD      HL,(FSIZE)      ; size by 128 bytes.
284:         LD      B,0
285:         ADD     HL,HL
286:         RL      B
287:         LD      C,H
288:         LD      A,L
289:         AND     A
290:         RET     Z               ; return if AMARI 0.
291:
292:         INC     BC
293:         RET
294:
295: ;
296: ; FILE RECEIVE
297: ;
298: FILINP:
299:         LD      A,16
300:         LD      (FUNCNO),A
301:         CALL    NMIINT
302:
303:         LD      A,(PARAM1)
304:         AND     A
305:         RET     Z               ; return, if not error.
306:
307:         CALL    DELETE
308:         JP      0
309:
310: ;
311: ; FILE TRANSMIT
312: ;
313: FILOUT:
314:         PUSH    DE
315:         EX      DE,HL           ; HL: file end address.
316:         LD      DE,BUFFER
317:         AND     A
318:         SBC     HL,DE
319:         LD      (FSIZE),HL
320:         POP     DE
321:
322:         LD      A,17
323:         LD      (FUNCNO),A
324:         CALL    NMIINT
325:
326:         LD      A,(PARAM1)
327:         AND     A
328:         JP      NZ,0            ; jump, if error.
329:
330:         RET
331:
332: ; ********************
333: ; *   nmiint         *
334: ; ********************
335: NMIINT:
336:         LD      HL,(NMIADRS)
337:         LD      (INTBUF),HL
338:         LD      HL,45EDH                ; RETN code.
339:         LD      (NMIADRS),HL
340:
341:         HALT
342:
343:         NOP
344:         LD      HL,(INTBUF)
345:         LD      (NMIADRS),HL
346:         RET
347:
348: ;
349: ;
350: ; CONSOLE INPUT
351: ;
352: KEY:
353:         PUSH    BC
354:         PUSH    DE
355:         PUSH    HL
356:         LD      C,1
357:         CALL    BDOS
358:         CALL    CAPS
359:         POP     HL
360:         POP     DE
361:         POP     BC
362:         RET
363:
364: ;
365: ; SET CAP. SHIFT ON
366: ;
367: CAPS:
368:         CP      61H
369:         RET     C
370:
371:         CP      7BH
372:         RET     NC
373:
374:         AND     5FH
375:         RET
376:
377: ;
378: ; CONSOLE OUTPUT
379: ;
380: TYPE:
381:         PUSH    AF
382:         PUSH    BC
383:         PUSH    DE
384:         PUSH    HL
385:         LD      C,2
386:         LD      E,A
387:         CALL    BDOS
388:         POP     HL
389:         POP     DE
390:         POP     BC
391:         POP     AF
392:         RET
393:
394: ;
395: ; FILE OPEN
396: ;
397: OPEN:
398:         PUSH    BC
399:         PUSH    DE
400:         PUSH    HL
401:         LD      C,15
402:         LD      DE,FCB
403:         CALL    BDOS
404:         POP     HL
405:         POP     DE
406:         POP     BC
407:         RET
408:
409: ;
```

▶きっと謎の福袋のなかにはビルディング・ホッパーが入っているに違いない。それとタイムシークレットの続編の3も入っていることだろう，と期待している私は受験生。
櫛渕　健一　(18)　神奈川県

```
410: ; CLOSE FILE
411: ;
412: CLOSE:
413:         PUSH    BC
414:         PUSH    DE
415:         PUSH    HL
416:         LD      C,16
417:         LD      DE,FCB
418:         CALL    BDOS
419:         POP     HL
420:         POP     DE
421:         POP     BC
422:         RET
423:
424: ;
425: ; DELETE FILE
426: ;
427: DELETE:
428:         PUSH    BC
429:         PUSH    DE
430:         PUSH    HL
431:         LD      C,19
432:         LD      DE,FCB
433:         CALL    BDOS
434:         POP     HL
435:         POP     DE
436:         POP     BC
437:         RET
438:
439: ;
440: ; READ SEQUENTIAL FILE
441: ;
442: READSQ:
443:         PUSH    BC
444:         PUSH    DE
445:         PUSH    HL
446:         LD      C,20
447:         LD      DE,FCB
448:         CALL    BDOS
449:         POP     HL
450:         POP     DE
451:         POP     BC
452:         RET
453:
454: ;
455: ; WRITE SEQUENTIAL FILE
456: ;
457: WRITSQ:
458:         PUSH    BC
459:         PUSH    DE
460:         PUSH    HL
461:         LD      C,21
462:         LD      DE,FCB
463:         CALL    BDOS
464:         POP     HL
465:         POP     DE
466:         POP     BC
467:         RET
468:
469: ;
470: ; MAKE FILE
471: ;
472: MAKE:
473:         PUSH    BC
474:         PUSH    DE
475:         PUSH    HL
476:         LD      C,22
477:         LD      DE,FCB
478:         CALL    BDOS
479:         POP     HL
480:         POP     DE
481:         POP     BC
482:         RET
483:
484: ;
485: ; CR,LF
486: ;
487: CRLF:
488:         PUSH    AF
489:         LD      A,CR
490:         CALL    TYPE
491:         LD      A,LF
492:         CALL    TYPE
493:         POP     AF
494:         RET
495:
496: ;
497: ; PRINT MESSAGE
498: ;
499: MESSAG:
500:         PUSH    BC
501:         PUSH    DE
502:         PUSH    HL
503:         LD      C,9
504:         CALL    BDOS
505:         POP     HL
506:         POP     DE
507:         POP     BC
508:         RET
509:
510: ;
511: ; READ CONSOLE BUFFER
512: ;
513: GETLIN:
514:         LD      A,(OPTFLG)
515:         AND     A
516:         RET     NZ                              ; return, if option.
517:
518:         PUSH    BC
519:         PUSH    DE
520:         PUSH    HL
521:         LD      B,40
522:         LD      HL,FILBUF
523: GTLIN0:
524:         LD      (HL),0
525:         INC     HL
526:         DJNZ    GTLIN0
527:
528:         LD      A,38            ; max 38 character.
529:         LD      (FILBUF),A
530:         LD      C,10
531:         LD      DE,FILBUF
532:         CALL    BDOS
533:         POP     HL
534:         POP     DE
535:         POP     BC
536:         RET
537:
538: ;
539: ; GET FILE NAME
540: ;
541: GETNAM:
542:         LD      B,30
543:         LD      DE,FNAME
544:         LD      HL,FILBUF+2
545: GETNAM0:
546:         LD      A,(HL)
547:         INC     HL
548:         CALL    CAPS
549:         LD      (DE),A
550:         INC     DE
551:         DJNZ    GETNAM0
552:
553:         RET
554:
555: ;
556: ; GET FILE NAME FROM CONSOLE
557: ;
```

```
558: FILNAM:
559:         CALL    GETLIN
560:         CALL    GETNAM
561:
562:         LD      B,40
563:         LD      HL,FCB
564: FILNM0:
565:         LD      (HL),0
566:         INC     HL
567:         DJNZ    FILNM0
568:
569:         LD      B,11
570:         LD      HL,FCB+1
571: FILNM1:
572:         LD      (HL),' '
573:         INC     HL
574:         DJNZ    FILNM1
575:
576:         LD      B,8
577:         LD      DE,FCB+1
578:         LD      HL,FILBUF+2
579: FILNM2:
580:         LD      A,(HL)
581:         INC     HL
582:         CALL    CAPS
583:         CP      '.'
584:         JR      Z,FILNM3
585:
586:         LD      (DE),A
587:         INC     DE
588:         DJNZ    FILNM2
589:
590: FILNM3:
591:         LD      B,3
592:         LD      DE,FCB+9
593: FILNM4:
594:         LD      A,(HL)
595:         INC     HL
596:         CALL    CAPS
597:         CP      CR
598:         JR      Z,FILNM5
599:
600:         LD      (DE),A
601:         INC     DE
602:         DJNZ    FILNM4
603:
604: FILNM5:
605:         XOR     A
606:         RET
607:
608: ;
609: ; OPEN ERROR
610: ;
611: OPNERR:
612:         LD      DE,DOPEN
613:         CALL    MESSAG
614:         LD      A,0FFH
615:         JP      0
616:
617: ;
618: FNERR:
619:         LD      DE,DFNERR
620:         CALL    MESSAG
621:         LD      A,0FFH
622:         JP      0
623:
624: ;
625: DISKER:
626:         LD      DE,DERROR
627:         CALL    MESSAG
628:         CALL    DELETE
629:         LD      A,0FFH
630:         JP      0
631:
632: ;
633: DEX:
634:         DEFB    CR,LF
635:         DEFB    ' *** 64180 <--> X68000  Ver 1.1 ***$'
636:
637: ;
638: DCOMND:
639:         DEFB    CR,LF
640:         DEFB    ' ** EX COMMAND TABLE **',CR,LF
641:         DEFB    CR,LF
642:         DEFB    ' E....ESCAPE TO CP/M   ',CR,LF
643:         DEFB    ' H....HELP             ',CR,LF
644:         DEFB    ' R....64180 <-- X68000 ',CR,LF
645:         DEFB    ' W....64180 --> X68000 ',CR,LF
646:         DEFB    CR,LF,'$'
647:
648: ;
649: DREADY:
650:         DEFB    CR,LF
651:         DEFB    '  Ready? (Y/N) $'
652:
653: ;
654: DREAD:
655:         DEFB    '  Read file name? : $'
656:
657: ;
658: DSAVE:
659:         DEFB    '  Save file name? : $'
660:
661: ;
662: DOK:
663:         DEFB    CR,LF
664:         DEFB    ' Complete'
665:         DEFB    CR,LF,'$'
666:
667: ;
668: DOPEN:
669:         DEFB    CR,LF
670:         DEFB    ' No file'
671:         DEFB    CR,LF,'$'
672:
673: ;
674: DERROR:
675:         DEFB    CR,LF
676:         DEFB    ' Disk error'
677:         DEFB    CR,LF,'$'
678:
679: ;
680: DFNERR:
681:         DEFB    CR,LF
682:         DEFB    ' File name error'
683:         DEFB    CR,LF,'$'
684:
685: ;
686: ;
687: ; WORK AREA
688: ;
689: OPTFLG:
690:         DEFS    1                       ; option flag.
691:
692: INTBUF:
693:         DEFS    2                       ; save buffer for NMI.
694: FCB:
695:         DEFS    40
696: FILBUF:
697:         DEFS    40
698: ;
699:         DEFS    256
700: STACK:
701:         DEFB    0E5H                    ; end mark.
702: ;
703: ;
704:         END
```

X68000のライバルは誰か?

ようこそ,セガ・メガドライブ!!

編集室
Saito Susumu
斎藤 晋

人気最高潮のパソコンX68000の前に,あまりにも挑戦的な価格で登場した68000マシン「メガドライブ」。体感ゲームでお馴染みのセガが世に送り出した強力な家庭用ゲーム機だ。しかも富士通からは32ビットのホビーパソコンが出るという。今,X68000は狙われている?

ライバル出現?

X68000という怪物のようなパソコンが登場したのが,1986年秋のエレクトロニクスショウであった。16ビットパソコンといっても,当時はPC-9801一辺倒で,CPUも,8086かせいぜい80286という時代である。

68000を載せたパソコンというだけで十分に話題となる時期ではあったが,X68000が熱狂的に迎えられた理由は単にハードウェアのスペックが飛び抜けていたからではないだろう。確かにそれはユーザーが待っていたマシンであったのだ。

さて,その後のX68000の展開はOh!Xを読んでいる皆さんならご承知のとおりである。オリジナルOSということで心配されたバグもなく,ソフトの充実度も進んで現在350タイトルのソフトが市場に出回っている。発売当初は懐疑的だった人々も安定した人気を認めざるを得ないようになってきた。価格的にはまだまだ高価なマシンでもあり,簡単に手を出せるものではないが,各雑誌の調査などでも次に購入したいパソコンのナンバーワンとなるケースも多いようだ。

しかし,こうした嬉しい状況が無条件に続くと思うのは間違いだろう。X68000ユーザーとしては決して慢心していられる状況というわけではない。

すでに,ご存じの皆さんも多いことと思うが,近く富士通から32ビットパソコンが発表となる予定である。32ビットといっても,これまでのFM Rシリーズの流れをくむビジネスパソコンではなく(だとしたらほとんど影響ないのだが),X68000と同様にパーソナルユーザーをターゲットにしたものということだ。

さらに,X68000にとっては,もっと気になる存在なのが,先ごろセガ・エンタープライゼスから発売された16ビットゲームマシン「メガドライブ」である。68000(8MHz)とZ80AをCPUに据え,スプライトやオリジナルサウンドチップなど強力なAV機能を備えながら,なんと21,000円というPCエンジンよりも安い価格を実現しているのである。

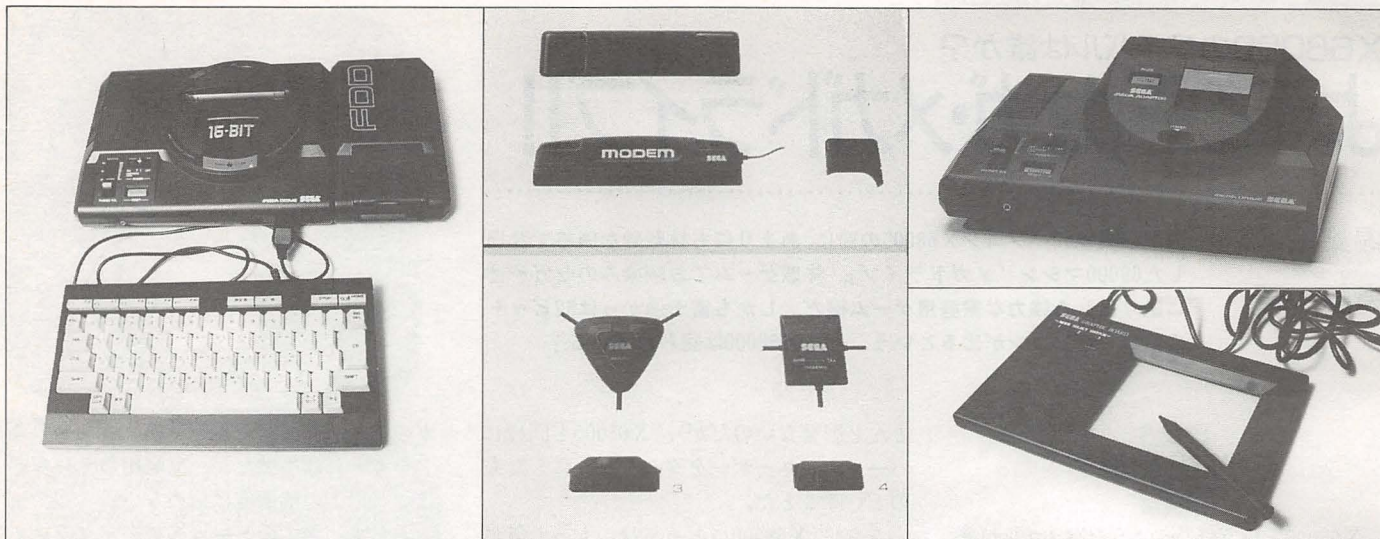
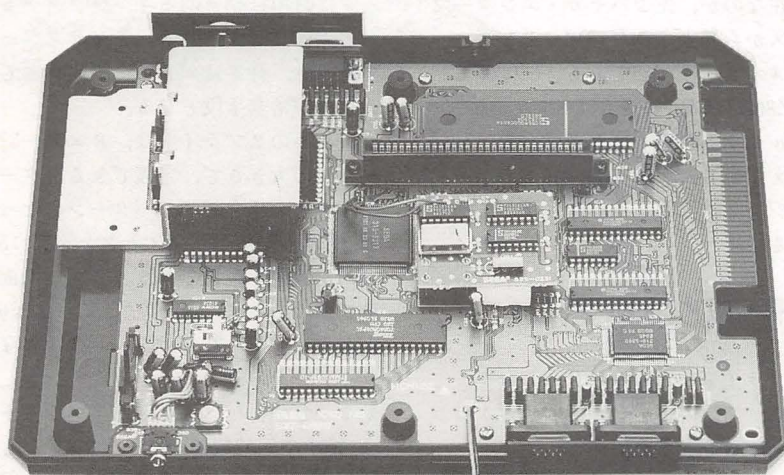
脅威のゲームマシン

メガドライブは基本的にゲームマシンであるから,当然のことながら勝負はグラフィックとサウンドである。

まずは基本的なグラフィック機能だが,色数は512色中から64色が選択できる。解像度は320×224ドットで,これはゲームセンターの業務用ゲーム機と基本的に同じ。セガのアーケードゲームを移植する場合のことを考えれば当然だが,家庭用のテレビではこれ以上の解像度は望めない。

そして,多くのゲームの場合,グラフィックは自由に動き回るキャラクターと背景とによって構成される。キャラクターはスプライト,背景はグラフィック画面で,というのが常套手段となる。

決め手のスプライトは,8×8~32×32ドットの大きさで,定義できるパターン数は8×8の場合で2048パターンとなっている。使える色はグラフィックと同様に512色中から64色が選択できる。一度に画面に表示できるのは80パターンで,これはX68000より少ないが,32×32ドットの大きなキャラクターを定義しても1パターンとして扱えるので強力だ。当初は,スプライトに拡大/縮小や回転機能があるのでは? との期待(恐れ?)もあったのだが,現在発表されているソフトを見る限り,そういった機能はなさそうだ。また,X68000のようにスプライトパターンで背景を作り,キャラク

これがゲームマシンの新星,メガドライブ(本体価格21,000円)。ファミコンやPCエンジンに比べるとちょっと大きいが性能は大きさ以上にすごいものがある。コントローラのデザインも斬新だ。同時発売のゲームカートリッジは3本で,いずれもセガブランド(各5,800円)。各地とも売り切れ店続出で品薄状態である。

メガドライブには2インチのフロッピーディスクドライブをはじめキーボード，モデム，タブレットを使ったお絵かきツールなども予定されている。また，右上の帽子のようなものはマスターシステムのソフトを動かすためのメガアダプターだ。

これがメガドライブ内部写真。基板は1枚で比較的ゆったりとした配置で石が並んでいる。右上のいちばん大きな石が68000，中央部やや左下にZ80が見える。カスタムチップは4つで，中央のクリスタルの下にバスコントローラ，左がグラフィックコントローラ，右下にI/Oコントローラ，そして左端の小さいのがサウンドチップである。また，右端のポートは来春発売予定のディスクユニットが接続できるようになっている。

X68000とメガドライブの主な仕様の比較

	メガドライブ	X68000ACE
CPU	68000(8MHz) Z80 (4MHz)	68000(10MHz)
メインメモリ	72Kバイト	1Mバイト(最大12Mバイト)
ビデオRAM	64Kバイト	512Kバイト(グラフィック) 512Kバイト(テキスト)
グラフィック	最大320×224ドット 512色中64色×2画面 ツインスムーススクロール 縦・横分割スクロール ウィンドウ機能	512×512ドット 65536色×1画面，256色×2画面 16色×4画面 1024×1024ドット 16色×1画面
スプライト	8×8〜32×32ピクセル 512色中64色，定義数2048 80パターン同時表示	16×16ピクセル 65536色中16色，最大定義数256 スプライトで背景を構成可能
サウンド	FM音源：6音(4オペレータ) PSG：3音+1ノイズ PCM	FM音源：8音(4オペレータ) AD PCM
価格	21,000円	319,800円

ターと重ね合わせるといった機能はない。

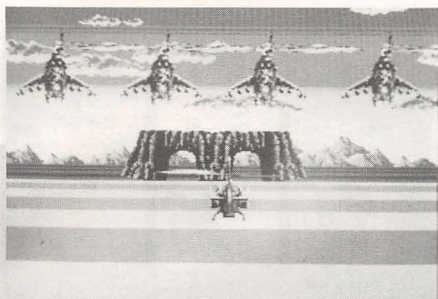
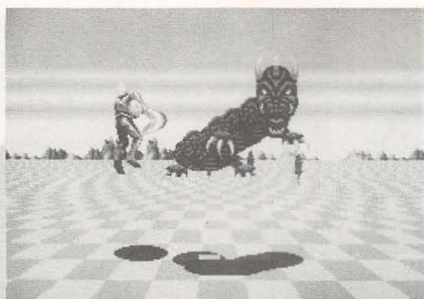
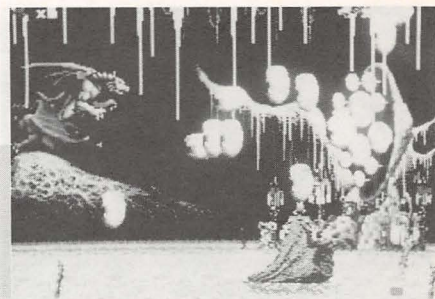
次に背景としてのグラフィック機能として有効なのが，ツインスクロールと呼ばれるもの。要するに，2画面のグラフィックを持っており，それが別々にスムーズスクロールできるという機能だ。このほか，ウィンドウを設定したり画面を分割してその中で画面をスクロールさせたりもできる。グラフィックに関してはさすがにX68000ほどにはいかないが（メモリ容量が圧倒的に違うから），ゲーム用としてはまずまずの機能といえるだろう。

次にサウンドに関しても，メガドライブにはヤマハとの共同開発によるオリジナルのサウンドチップを載せている。FM音源6音＋PSG3音（＋1ノイズ）で，おまけにPCMもあわせ持った，まさしくゲームのために作られたカスタムチップといった感じだ。

さて，これらはいわば表面的な機能だが，これだけを見ても「メガドライブ」がかなりのマシンであることがわかるはずだ。いくらゲーム専用に特化して不要な部分を切り詰めたからといって，本当に2万1,000円でできるのか不思議に思うことだろう。

メガドライブのハードウエア

というわけで，メガドライブのハードウェアについて見ていくことにしよう。

メガドライブの心臓はなんといってもMPU68000。CPUといわずMPUと呼びたくな

スーパーサンダーブレード（発売中）
戦闘ヘリ「スーパーサンダーブレード」の青い機体が華麗に宙を舞う飛行シミュレーション＆シューティングアクションだ。高層都市をかいくぐり，海上を越え，3Dステージから2Dステージへと果てしない攻防が繰り広げられる。

スペースハリアーII（発売中）
体感ゲームの名作「スペースハリアー」がさらにパワーアップ。新しいデザインのボスキャラが次々と登場。嬉しいことに，最終面以外の12面は面セレクトで好きなステージから始められるようになっている。

獣王記（発売中）
ゼウスの言葉が地に響き，伝説の獣人族の戦士は長い眠りから覚めた。アテナを救うため，獣戦士の戦いが始まる。3ボタンをフルに生かしたパンチ，キック，ジャンプのアクションで必殺技を繰り出そう。

るのは「68000は電卓あがりのどこぞの石などとは違うんじゃあ」という気持ちの表れだろうか。まあ，CPUでもMPUでも同じことなんですけどね。で，ここで使われている68000はMacintoshと同じ8MHz版で，最もオーソドックスなタイプ。ちなみに，X68000では10MHzでシュリンクパッケージのちょっと小型のタイプが使われている。

また，メガドライブにはもうひとつCPUが載っている。お馴染みのZ80Aだ。

いまのところZ80は主にサウンド関係のために使われているということだが，単なるサブCPUとしてしか使えないわけではない。68000とZ80は，お互いにバスを共有し，Z80は68000の持つアドレス空間のどこにでも配置できるようになっている。内部の写真を見ると，ちょうど真ん中あたりにいかにもパッチを当てたような小さな基板があるが，実はその下に，2つのCPUのバスをコントロールするためのカスタムチップが隠されている。また，右下のコントローラを接続するポートのそばにあるのがI/Oコントロール用チップ，これもカスタムだ。

そして，主力となるスプライトとグラフィックにも当然のことながら専用のカスタムLSIが採用されている。基板のほぼ中央にある正方形のチップがそれ。論理回路だけで3万ゲートもある強者である。

このように，主要な機能のほとんどをカスタム化することでこれだけのスペックを数個の石に詰め込んでいるわけだ。

しかし，一方でメガドライブのハードは開かれたものでもある。CPUまでもオリジナルに設計されたファミコンなどと違い，68000やZ80といった一般的な石を使い，しかもすべてのバスが開放されている。拡張性についても心配ないし，68000の持つ豊富な開発環境が利用できるのはソフトメーカーにとってありがたいことである。

肝心のソフトウェアだが

ハードウェアはこのくらいにして，ソフトウェアのほうに目を向けてみよう。メガドライブ本体と同時に発売されたのは，「スーパーサンダーブレード」，「スペースハリアーII」，「獣王記」の3本。メガドライブの性能を見るには，これらのソフトを見てみるのが早い。

Oh!X読者にとり最も馴染みの深いのは，やはりスペハリだろうか。今回，メガドライブ用に開発されたのは，かなりアレンジされた「II」であるが，ゲームの基本構成はやはりあのスペハリだ。X68000に移植された当時は「本当にできるなんて!?」と驚いたものだが，いともあっさりとやってくれるあたり，さすがはセガと唸らされる。しかも地面はしっかり格子状のパターンで，キャラクターの影もちゃんと映っている。

体感ゲームからもうひとつ，サンダーブレードにスーパーがついての登場だ。これもなかなかのシューティングゲームだが，セガの体感ゲームのなかでもグラフィックの微妙な色合いや遠近感などが大きくイメージを左右するタイプなので，色数やスプライトの機能で業務用マシンに劣るメガドライブではやや苦しいものがあるようだ。

そして，3本目の獣王記だが，これはメガドライブのツインスクロール機能がこれみよがしに見せつけられるゲームだ。グラフィックも色数を最大限に生かしており，

熱烈大歓迎のメガドライブ

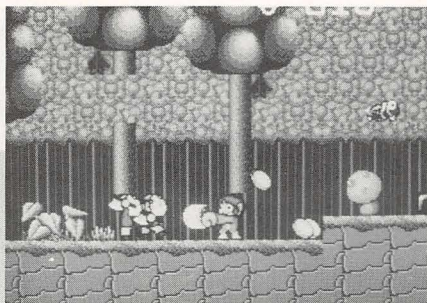
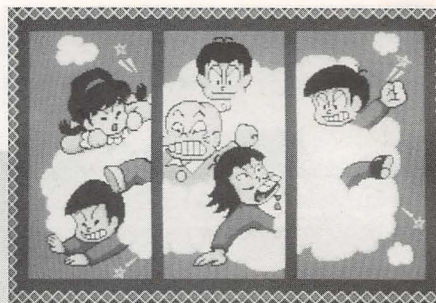
やりましたねえ，「オメガトライブ」じゃなかった「メガドライブ」。うーん，さすがはセガ。思ったとおり68000ですよ。

セガといえば，ハングオン，スペースハリアー，アウトラン，アフターバーナー，といった体感ゲームで，ゲームセンターをネクラな少年の隠れ家から，アミューズメントランドへと変身させたビデオゲームメーカーなわけですよ。最近は，ギャラクシーフォースでもって，グリングリン回りながら敵を撃ち落としちゃうんですから，そりゃもう恥ずかしいのを通り越して快感なんです。

で，そのセガのビデオゲームには，68000が多く使われてきたんですね。だから，X68000が出たときはみんな期待しちゃったわけですよ。もしかしたら，スペハリとかアウトランができるんじゃないかってね。もちろん，スペハリなんかは電波新聞社によって実際に移植されたわけ。ただしですよ，業務用の基板からキャラクタデータなどはもらえないことになっているので，移植といってもデータから作んなきゃならない。コナミのグラディウスだってビデオを見ながら一生懸命データを作っていったというんだから大変な労力ですよね(SPSさんご苦労さま。でもちょっと沙羅曼蛇は遅いよ)。

で，私が何を言いたいかわかります？　メガドライブ用にいろいろなゲームソフトが開発されるとしますね。するとメガドライブからX68000への移植という線だって考えられるわけですよ。スプライトをギンギンに使ったり画面を2重にスクロールさせたりしてもX68000になら簡単に移植が可能なはず。そして，メガドライブからならキャラクターデータをそのまま移すということも契約によっては可能なんじゃないかと思うんです。

もちろん，逆にX68000からメガドライブへ移植したっていいですね。いままで，X68000用のゲームソフトといえば，アーケードゲームからの移植ものが多いというのが特徴だったですね。パソコンのソフトハウスがX68000ならではのソフトを作ってしまうと他機種に移植しづらくなるからだと思うんですよ。それがメガドライブの存在があるとしたら，ソフトハウスだって安心してX68000用のソフトに開発費がかけられるんじゃないかと思ったりするんですよ。うーん，こりゃラッキー！　（謎の東洋人X）

ファンタシースターII（2月発売予定）
マザーブレインの力で緑の星となったモタビアに異変が起こり始めた。大発生した凶悪なバイオモンスターに対し，敢然と立ち向かうモタビアの戦士たちの物語。戦闘モードもメガドライブならではの迫力で楽しめるSFファンタジーRPGだ。

アレクの天空魔城（1月発売予定）
セガならではの明るくかわいいキャラ，アレックスキッドがメガドライブで大冒険。今回は，ジャンケン大王を倒したアレクが死んだはずの父を捜してジャンバリク星へと旅に出るというお話。楽しい乗り物や愉快な敵キャラも次々と登場。

おそ松くん（12月発売予定）
懐かしの「おそ松くん」がリバイバル人気で大活躍のおとぼけアクション。もちろん，おフランス帰りのイヤミや，チビ太，デカパン，ハタ坊，そしてトトコちゃんなど，レギュラーキャラが総登場でメガドラする。

スプライトの動きもダイナミックでよい。

このほか，アクションゲームでは，「スーパーハングオン」，「パワードリフト」，「アウトラン」などが，RPGでは期待度ナンバーワンの「ファンタシースターII」，シミュレーションではあの「SUPER大戦略」などがセガのブランドで予定されている。

そして気になる，サードパーティの参加だが，すでに大手ゲームメーカーのほとんどがメガドライブ用ゲームカートリッジの開発に乗り出すことを決めている。

X68000はどうなる

メガドライブが登場したことで，X68000の立場はどうなるだろうか。一部ではあい変わらずのことだが，X68000はゲームマシンとして見られることが多い。続々と登場するアクションゲームを見れば，そういう印象があってもしかたがないかもしれない。しかし，ゲームマシンと呼ばれる限り，メガドライブの存在はかなりの脅威となる。少なくともゲームにしか関心のない人は(これまでのX68000ユーザーは違うようだが)ムリをして高価なX68000を買わなくてもスペハリクラスのゲームをビシビシと楽しめるようになるからだ。なにしろ相手は21,000円，X68000ACEは15倍も高い319,800円だ。

かつて，ファミコンがブームを巻き起こしたころから，パソコンはそれまでの勢いを失っていったという歴史をもつ。ゲーム志向の強いユーザーがパソコン離れを起こしたのだという見方が一般的だ。今また，メガドライブがX68000の神話を崩すのではないかという不安が生じるとしてもしかたがない。

しかし，一方で別の見方もある。かつてパソコンがファミコンに食われたのは，単にファミコンが価格的に安かったからというだけでなく，パソコンの機能自体が低すぎたのだ。実際には，ゲームをするための機能だけをとっても，パソコンよりもファミコンのほうが上回っていたのである。これではパソコンに勝ち目はない。

X68000は発表から2年たった今でもユーザーにとって憧れのマシンである。そしてグラフィック，サウンド，ビジネスのあらゆる用途に対応できる十分な機能を持ち，なおかつゲームマシンとしても，まだメガドライブにとって代わられるものではないだろう。むしろ，同じ68000ファミリーとして歓迎されるべき相手であるはずだ。

X68000にとって，メガドライブが脅威となるか，あるいは強力な味方となるかは，本当のところX68000ユーザー全体の自覚にかかっているのである。

*　　*　　*

P.S. かねてから噂のファミコンの16ビット版だが，このほど任天堂から公式な発表があった。名称はこれまでいわれてきたとおり「スーパーファミコン」だが，発売はなんと来年，1989年の7月ということだ。これは「先手必勝の構えで登場したセガのメガドライブに対し，それまで準備してきたものでは太刀打ちできないと見て急遽仕様を変更したのでは？」というのが，大方の見方である。当然，発表された仕様はメガドライブを上回るものだが，ハードはおそらくこれから本格的な開発に入るのだろう。いずれにしても玩具の発表としては，鬼が笑うような話である。

果たして富士通の切り札は？

「電脳遊園地」in東京ドームで公開されるはずだった富士通の32ビットホビーパソコンは，イベントが開催延期になったために公開もおあずけをくらったかたちとなっている。

ここでちょっと，関連する記事の載った他誌の情報を拾ってみよう。

日経パソコン(11月14日号)によると，CPUは80386でMS-DOSを採用，価格は40万円程度でゲーム用に高いAV機能を持ち，発売と同時に新作ゲームソフト100本を用意する，となっている。情報のいいかげんさもすごいが，新作ゲームソフト100本というのは，かなり限度を超えていて面白い。

ところが，ASAHIパソコン(12月1日号)を見るともっとすごい。「来年早々，32ビット新機種とソフト500本を発表」とある。ここまでくると，もうなにがなんだかわからない。とりあえず，「日経パソコンの100本に対し，ASAHIパソコン500本で断然有利！」ということでこの話題はここまでにしよう。

ともかく，32ビットというからにはCPUは，80386か68020または68030のはず。富士通のマシンといったらやはりグラフィックとサウンドだ。16777216色同時表示ぐらいは狙ってくるだろう。ビットマップ用の専用コントローラや拡大縮小回転のできるスプライトが載っていてもおかしくない。もっとも，X68000の2年も後を追うマシンとしてはそれぐらいはやらないとまともな勝負にはならないだろう。もしも68030だったら，これはちょっと怖い。まして，OSがUNIXだったりすると……，機能はともかく，ユーザーの期待度はかなりのものになるはずだ。逆に80386であればそれほど心配することはないかもしれない。

例は悪いが，PC-88VAがほとんど敗北したのは，単に機能が低かったからではない。VAにはX68000の対抗機種として価格やスペックが設定されていたようだが，そこにはもっと基本的な勘違いがあったようだ。結局のところVAはユーザーの心をつかむことに失敗したのだから。

富士通が起死回生のニューマシンを投入してくるとすれば，ハードウェアは申し分なくX68000を上回るものとなるだろう。果たしてそれがユーザーの期待度をかきたてるようなものか，あるいは88VA以上の巨大な勘違いとなるか。今後の発表に注目したい。（T）

第31回

猫とコンピュータ

ちょっと宇宙人

Takazawa Kyoko
高沢 恭子

今回は座談会にも出席してますます忙しそうな恭子さん。分野の異なる人たちとの語らいはぜひ必要なものですが，そこでギャップが生じたりすると，フォローするのはなかなかにたいへんなのです。

パソコン通信局FBIのネット上で「X68000を購入したいのですが，お使いの方は感想をお聞かせください」とお願いしたら，何人かのお友だちが答えてくださった。

「ううん　私もX68000ほしいです」というタイトルで「ITMOMO」さんから，「けっこう初期投資もかかるし，ランニングコストも相当かかりそうで二の足を踏んでますが……買うならやっぱり98かな（うっ，悲しいさがですね)」。

「みゆ」君は「私，使ってます」のタイトルで，「使用感と言われても，仕事に使っているようないないような……エディタなども少ないし，FEPもだんだん出てきているようですけど，これだっていうものがないです」

FEPというのはフロントエンドプロセッサ，つまり日本語入力などの，OSにない機能を実現するものだ。

「MINE」氏は「X68000はソフトがなかなか数が少ないのでは……たとえば，CGをやるにしても，どのソフトがいいかほんとに情報が少なくて」

「X68000ですかぁ」と「ちなみ」さん（青年です)は，「あれはソフトがあんまり出ていないのでは……モノクロでよければMacをおすすめしますよ。CGだったら下はマックペイント，上はイラストレータ'88，フリーハンド，などなどソフトに不足はありません。それに操作性も抜群ですしね」

そして親愛なる「ちゃがま」会長は（FBIには「たかざわきょうこファンクラブ」というボードがあるんです。彼は頼りになる会長さんです)，「ふーむ，私はお金がないから逃げてるだけで，お金があったらたぶん買ってますからねぇ。どうなんでしょ？確かにソフトは少ないし，高いよーな気がします。ハードを買ったがためにただでさえ少ないソフトが買えない，とかねぇ」

筆とマウス

X68000を買ったら，まずZ'sSTAFF（そのあとにPRO-68Kなどと付くらしいけれど）というソフトで絵を描いてみようと思っている。それが当面の目的だけれど，果たして後悔しないかしらとずーっと考えてきた。

正直にいって，機械で描く絵にまだまだ疑問や，ささやかな反感を持っている。あの緻密さや硬さや，異様な発色の良さがすぐに思い出されて，筆の持つ粗さや柔らかさが浮かんでこないのだ。描画より写真の確かさに近いようで，不確かな線や筆勢などを表現するにはふさわしくないような印象を持ってしまう。

だから驚きの声を上げるような，現実を超えた密度と輝きのある画面を現出しながら，なんだか個性のありかがわからないのではないかしらという不満がある。

でもそれは私の偏見かもしれない。Z'sSTAFFの機能の概要に目を通すと，およそこちらで期待している一切の技法が実現できるらしいのだ。

「ペン」の選択であらゆるタッチが表現でき，「パレット」で自由自在に色を選んだり混ぜあわせたりもできる。ぼかしたりにじませたりの描法も可能だし，フリーハンドの線も幾何学的な線も描ける。「タイル」や「スクリーントーン」などというパターン表現もある。

これにコンピュータならではの編集の機能が加わると，とても人為では不可能な手品のようなワザが実現できるようだ。

実際に絵の具を混ぜあわせてキャンバスに塗る，気に入らない箇所を布でふきとったり，ナイフで削ったりするというのも，楽しみはあるけれど苦労なものだ。パソコンならファウンデーション（地塗りなどの基礎）の手間もいらないし，使い終わったパレットや筆の後始末もいらない。第一パソコンの前に椅子ひとつだけのスペースがあればよい。しかも3万色以上の表示ができるなんて。手持ちの絵の具でそれだけの表現をしようとしたらたいへんだろう（特に油絵の具は1本がわりと高価なのだ)。

要は自分の創造性や描く力量の問題で，まずは試してみなくては始まらない。しかし，あの，機械による表現の差はやはり気になってしまう。

ともかく，次の休日あたりに，いよいよX68000をもとめに出かけてみよう。

ウイケットの頭巾

FBI（FORESIGHT BULLETINBOARD INFORMATION SYSTEM)-NETは開局4年目を迎えた。カワムラさんのホストプログラム「BIG-MODEL」は，バージョンアップにつぐバージョンアップでますます充実，会員も500名を突破した。あいかわらずの無料ネットで，本業を別に持ちながら，24時間，4回線を維持するシスオペ，ナカムラさんのご苦労にはまったく頭が下がる。

「お母さん，ホンニャアどこ？」

ドイ君とふたり，2階からかけおりてきてトオルが聞いた。

「パパの部屋にいたわよ」と私。

パパの部屋＝マシンルームである。

「やった！　お母さん，このくらいのキレない？」

「あるけど……」何に使うの？と聞く必要はない。あれだな。

ジョージ・ルーカスの「スターウォーズ3」がテレビで初放映されてからというもの，トオルはあの「イウォーク族」の子供「ウィケット」が大好きになった。

ドイ君も中学校の視聴覚室で，担任の先生やお友だちとビデオで全編を見てすっか

りウィケット坊やのとりこになったのだそうだ。猫とタヌキと猿をミックスしたような愛くるしさと，あのリラックスした頭巾のファッション。ポシェットみたいな肩かけカバン。赤褐色の丸い目と原始的でチャーミングな言葉。

きのうは，FBIの「うさぎ組」のお友だちからいただいた「開運まねき猫」のぬいぐるみに，紙袋で作った頭巾がちゃんとかぶせてあった。

きょう，ふたりでホンニャアをつかまえて何をやろうとしているかはもう歴然だ。両耳と顔の部分がのぞくように穴があき，肩まですっぽりおおわれるあのイウォーク族の頭巾をホンニャアにもかぶせてみようというのだ。

引き出しの中からベージュ色のハギレを選び出すと，ふたりはさっそくハサミを入れようとしている。

「ねえねえ，思っているより難しいのよ。猫の頭をちゃんと立体的に考えて，耳の位置と目鼻の位置の関係をもう一度確かめないと……」

しまった，また私がいちばん積極的になろうとしている。

電話が鳴ったために頭巾作りへの参加は思いとどまることができた。FBIのメンバーで経済誌のジャーナリスト「IB（アイビー）-PATA（パタ）」(またの名を「松風散人（しょうふうさんじん）」)さんからだった。

「あのね，日経新聞の座談会に出席してみませんか？できたらキョウコさんの連絡先を教えてほしいと言われているんだけど」

「えーっ！　テーマは何ですか？」

「詳しいことはわからないけど，パソコン通信について話すことになるらしいですよ」

「わぁ，だいじょうぶかしら……」

「TRY，TRY，やってみなくちゃ」

「うーん」と数秒考えて，「じゃ，お願いします」

「OK！　それでなくっちゃ。たぶんこの電話のあとにTPOのキクザワさんという人から連絡があると思いますよ」

通信は私がパソコンに親しむ大きな役割をしてくれたし，今後も日常生活との関連や実用の面でいろいろな可能性を期待させてくれている。連日，新聞紙上でも各方面からの注目が盛んだし，こんな機会を与えてくれるのもネットで得た人間関係のおかげである。PATAさんありがとう。ただし，こういうふうに急を要するときの連絡はパソコン通信より電話のほうが当てになるというのがちょっと皮肉だ。

がんばらなくちゃ

キクザワさんという人からの連絡で，日経新聞の夕刊に掲載されるという座談会のおよその様子がわかった。毎回テーマを変えて著名人の司会で進められるもので，今回のテーマは「ホビー」，司会はイラストレーターで自らはソウルミュージックの収集が趣味というナガイヒロシさん。

私は「パソコン通信」のホビイストということらしい。ほかの出席者は「ホームビデオ」の外人男性と，「オートバイ」の若い女性だという。

通信は私にとって趣味であることにまちがいはないけれど，現在では生活習慣の一部という感じで，むやみにアクセスすることもあまりない。原稿を送るにもFMネットを利用したり実用にもなっている。でも座談会などというのは初めての経験だし，よい勉強になるだろう。私の期待と緊張は一気に高まった。

ひとつの分野からひとりとなると，責任もあるし誤ったことは発言できない。でもパソコン通信の，今までとまったく違う新しいコミュニケーションの便利さと合理性，開拓しだいで広がると思われる活用範囲の大きさ，しかも私をもって示す習得のたやすさ，友人との交流が生む利益の数々，そんなものをゆがめずに伝えたい。

使命感の強さから，興奮はなお増してくる。自分が始めたきっかけ，それ以前の少し歴史的なわが家の環境やパソコンライフの立地条件，パソコンクラブFORESIGHTの強力なメンバーの存在と草分け的なBBSネットの開局。ネットへの参加のチャンスを与えられて急速に増したパソコンへの親しみ，マシンを操れたことの喜びと斬新な世界の発見，空間を超えたスクランブルの交流。強制されずに個々に選べる自分とネットとの関わり方，そして実例としてのFBIネットの階層構造，メンバーの多彩さや活動例，etc. etc.……。

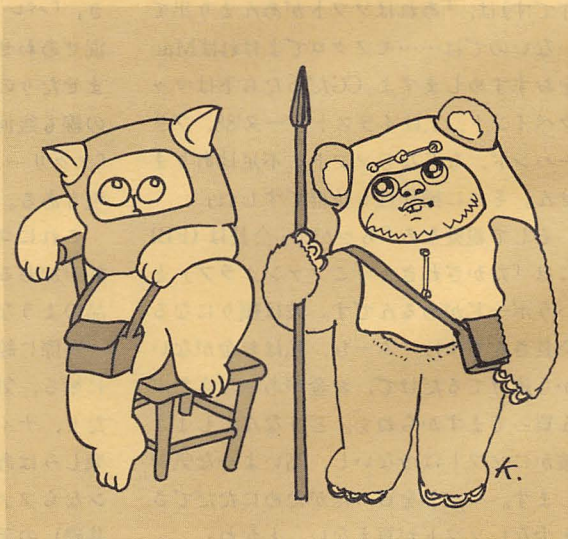
だけど，座談会は講演会じゃない。しかも1時間ちょっとの予定と言ってたっけ。ほかの人たちはみんな違う分野だし，個々のホビーというより，生活の中での「趣味」の価値を語りあうことが目的であるのは目に見えている。しかし「通信」の実態を示さずに自分の中での価値は説明できない。登場したばかりのまだ知られていない分野を，どうやって4分の1の発言の中でできるだろう。その日は刻々と近づいてくる。

原宿へ

「原宿のナガイさんの仕事場に，午後7時までにおこしください。30分ほど写真撮影して，そのあと同じビルの地下1階のクラブで，お食事をしながらお話ししていただきます」

私は7時10分前くらいに，出席者としては1番目の到着だった。ナガイさんはおヒゲをはやされた明るい率直な方で，40歳になられるそうだ。部屋の中ではすでに座談会の構成をするプロダクションの方やカメラマンの人たちが準備を進めていた。

やがて，きょうの記事をまとめるフリーの女性記者が，そのあと日経新聞の担当者が到着。「ホームビデオ」のロバート氏と「オートバイ」のダイゴクミコさんが揃ったのは7時をちょっと過ぎてからだった。

撮影は，ナガイさんのソウルミュージックのコレクション，LP盤8000枚，シングル盤1万枚というレコードをびっしり収めた，天井まで届く作り付けの棚の前で行われた。

座談会場は確かにクラブのようで，照明はあくまでも暗く，ガラス張りの向こうの吹きぬけの階下では，ミラーボールといっしょにかなりの音量でディスコ音楽が鳴り響いている。一隅の流線型のテーブルを囲んで，ここでまぎれもなく「座談会」をやるというので，少なからずびっくりしたけれど，お忙しいナガイさんの仕事場から最も近い場所がここだったのだろう。

バイキング式の料理がつぎつぎ運ばれてきて，出席者のほかに4，5人の関係者も座につき，いよいよ語りあいということになった。

アセっちゃダメなのに！

ナガイさんはグラフィックデザイナーからイラストレーターになられたそうで，「趣味」のソウルミュージックも，音楽番組の選曲を担当されるほどの本格派らしい。

「ホームビデオ」がホビーという，ロバート・レッド・ベアさんは千葉県にある女子短大の英文科の講師で44歳，赤毛のヘアとおヒゲから，レッド・ベアは半ばニックネームなのだそうだ。ビデオは広島国際アマチュア映画祭で総理大臣賞をとるなど，受賞は数知れぬほどだそうで，ビデオの撮影は注文を受けて出張もしているというから，「趣味」を越えている。日本での生活はもう20年近いそうだ。

オートバイのダイゴさんは28歳で，スラリとしたなかなか美しい人だ。本業は主に映画の紹介記事を書くフリーライター，バイクは免許を取ってまだ半年なので自分の車はこれから買うそうだ。

足元から響いてくる音楽は気になるけれど，ナガイさんのくったくのない語りかけで少しずつ会話が始まった。

ロバートさんはもともとハワイの大学で演劇を専攻，表現することが大好きなのでビデオもみるみる実力を上げていったようだ。注文されて撮影に出かけるのは，責任という怖さがあって，「趣味」と「遊び」とは別のものだという感じがするそうだ。

ダイゴさんは仕事の性質から，試写会の会場や会社の中などの室内で過ごすことがほとんどなので，気分をリフレッシュさせるためにバイクを思いついたのだそうだ。

さて私は……となると。ナガイさんは司会者の立場と個人的な素朴なお気持ちの両方を含めて，こうおっしゃった。

「パソコン通信ってなんなの？　これがわからないと話にならなくってね」

きっと，対談や座談会ではとても基本的な大事な質問なのだろう。ありがたいと思いながらも，ほかのおふたりにはこの質問がなかったことの違いを実感した。

実際にナガイさんはメカ嫌いだというし，ダイゴさんも，そのほかの関係者たちも，みんなパソコンには触らない人ばかりなのだ。唯一，ロバートさんが1年半ほど前に中古の98を買ってそのままになっていると言っていた。

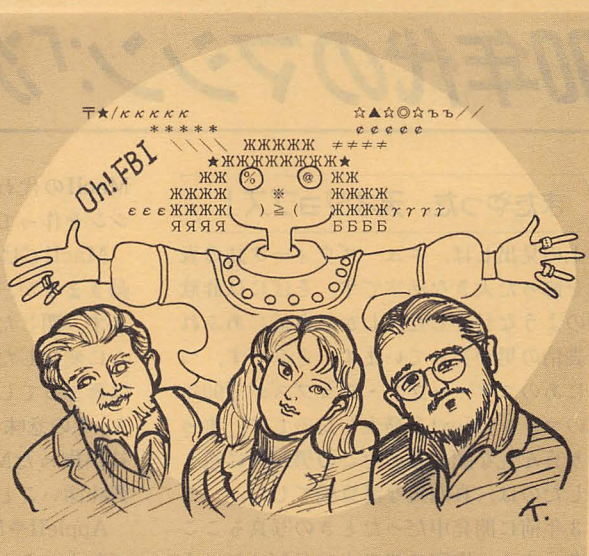
「まずパソコンの端末機に…つまりキーボードのついているアレですね…モデムというパソコンの信号を電話の信号に置き換える機械をつなぎまして，それを電話線につなぎ，電話回線を利用して……」

言いながら，きょう自分が胸の中に抱えてきたたくさんの荷物が，途方もなく大きすぎたように感じはじめた。ビデオやバイクにはすでに無用の「イントロダクション」のために，きっときょうの私は時間を使い果たしてしまうのだろうな，そんな焦りが湧いてきた。

ひとつのしくじり

ロバートさんは日本語については，聞くことはだいたいわかるけれど，話すほうはもうひと息なのだと言っておられた。確かに，時折なかなか言葉が見つからないふうで，私たちもいっしょに努力したが，お互いに少しずつもどかしかった。

ダイゴさんも物静かな口数の少ない方で問われることに微笑みながら答えている。

そんな中で私は，伝えたいことのあれこれが頭いっぱいポップコーンみたいにはじけていた。だから，にぎやかな音楽に負けまいということもあるけれど，ナガイさんやほかのふたりに比べると声がいちだんと大きくなる。自分でもそんな様子がひどくおかしい。私が話し始めると，みんながよその星の話を聞くような顔をしているみたいだ。マニアの仲間にいたらシロウトそのものの私が，いつから宇宙人になったのかしら。

「アクセス」，「ボード」，「階層」，出てくる用語にすべて説明が加わるような話だから聞いてくれる人もよほどの誠意がなくてはならない。でも音楽にさえぎられてどこまで伝わっているかもさだかでない。ひとつのことがらを語りきれないうちに，脈絡もなくほかのことを語らなければならなくなる。でも会話とはそういうものだから仕方がない。

テーブルの上では，飲み物や料理の間に置かれた小さなラジカセがひたすら回り続けている。この録音テープを起こし，じょうずに拾い集めた文章によって，「座談会」は難なくできあがることだろう。私は何も案ずることはないのだ。

ナガイさんは芸術家らしく，終始率直で聡明な方だった。でもその率直さで最後にこう言われたのが，私のこの日の失敗を象徴したかのようだった。

「僕はソウルミュージックの仲間がたくさんいるし，よく集まるけれど，そういう場所へは，そのジャンル以外の人は来ないから安心できる。でも通信ってのは，ワケのわからないヤツでもなんでも集まってくるみたいだなぁ，そんなのイヤだなぁ」

本来パソコン通信は，そういった点での選択を最も得意としているはずなのだ。自分の体験からもパソコンへの偏見を取る手段のひとつとして，なによりも強力なパソコンの実用例として，もっと「通信」を正しく紹介したかったのに，私はやっぱり気負いすぎてしくじったらしい。

90年代のマシン:「次」は「Next」に決まり!

またやった，天才ジョブズ！

上の見出しは，ニューズウィーク誌の表紙を飾った大きな活字です。そばには詐欺師のような目をした，しかし自信にあふれた表情の男が笑っています。そうです，ついにあのスティーブン・ジョブズが「90年代のマシン」Nextを発表したのです。すっかり真ん丸な顔になったこの男が世に送り出したのは，真四角な計算機でした。

3年前に開発中だったときの写真もここにあります。窓際の机の上にはMacintoshが置かれており，彼がジーパン姿でこちらを見ています(このころは痩せている)。当時，Next社の社員はたった10人だということや，読者への“I am still alive！”というメッセージを見て，内心アーアとため息をついたものでした。しかしその彼が，ついに世界中の期待に応えるべく再登場したのです。

スティーブン・ジョブズ。パーソナルコンピュータの世界に影響を与え続けてきた男。パソコンを常に真にパーソナルなものにし続けてきた人物。彼はもうひとりのスティーブ（確か万能リモコンかなにかを作ったというニュースは記憶している）とApple社を作り，あの懐かしきAppleIIを売り出したのです。パソコンの創世紀です。そしてMacをこの世に送り出しました。もちろん，失敗作といわれているAppleIIIとLisaを作ってきたのもそうです。

しかし，ジョブズはMacIIは作りませんでした。Apple社の経営陣に迎えたスカリーに逆にクビにされたからです。その辺りの事情には僕はあまり詳しくありませんが，MacIIの代わりにNextという90年代のマシンを作ってくれたわけです。

MacIIでは，一応Macの素直なソフトは動きますが，設計思想は大幅に変化しています。閉じたシステムから開かれたシステムに変化したことにより，独自の設計思想は薄まってしまったといえるかもしれません。その意味では，ジョブズのいないApple社が作ったMacIIとNextを比較するのは興味深いことでしょう。

AppleIIやMac（現在の冷たい色のものでなくもう少し暖かい色の時代のもの）に対する懐かしさや愛着を，なぜか最近のApple社の製品であるMacIIやMac SEに対しては持てなくなってきた原因がようやくわかったような気がします。「テクノパンク」ジョブズが作ったApple社は，ガレージの中でパソコンを作っていた古き良き時代の香りを失ってきたのかもしれません。会社の成長とは結局そんなものなのでしょうか。

図1　Nextのハンドルソフトウェア(予定)

システムソフト
- Mach, Display Post Script

開発ツール
- GNU C, GNU EMACS, Objective-C, Information Builder

オブジェクトオリエンテッドキット
- Application Kit, Sound Kit, Music Kit

データベース用アプリケーション
- Merriam Webstar & Ninth New Collageate Dictionary,
- Shakespeare: The Complete Works

アプリケーション
- Personal text database
- Electronic mail application
- Wordprocessor “Write Now！”

大きな声で「90年代のマシン」

ひと目見てユニークなのはNextの外観です。真っ黒な1フィート角の立方体で，キューブとも呼ばれています。また，本体には一度も手を触れずに起動からシャットダウンまでできるので，実際好きなところに転がしておいても平気です。

ロゴマークにもお金をかけており，IBMのあの有名なマークを作ったデザイナー，ポール・ランドに10万ドル(約1,200万円)を払って制作させました。さほど奇をてらったものではなく，キューブを意識したすっきりしたマークになっています。

さて，Nextの肝心のハードウェアについて見てみましょう。まず，CPUまわりの基本的な枠組みについていえることは，現在選択できるスペックとしてベストに近いものを選んでいるということです。人によってはオリジナリティがないというかもしれませんが，SUNのようなメーカーを除けば，CPUメーカー以外が自分で実用的なCPUを開発することは，マーケティングの面からも技術の面からも容易ではない，ということを忘れてはなりません。

CPUは68030(25MHz)，コプロセッサとして68882(25MHz)，メインメモリは8Mバイト（さらに8Mバイト増設可）です。参考文献3)によると，68030のバーストリードサイクルを可能な限り利用しており，このモードでは128ビット/クロックサイクルのメモリとCPU間のデータ転送速度を実現しているとか。これは通常のモードの2倍近い速度だそうです。

入出力関係に関しては，特に90年代のパーソナルコンピュータとしての模範となるべき条件をいろいろな面で満たしていると思います。まず第一に，標準装備の読み書き可能な光磁気ディスクを挙げなくてはなりません。容量は256Mバイトに及びます。その信頼性についてはデータが手元にありませんが，たぶんハードディスクなどのメディアよりも上だと思われます。

さらに，このマシンにはぜいたくにもデジタルシグナルプロセッサ（DSP）が装備されており，種々の信号処理，音声処理，数値処理などでの活躍が期待されます。音声に関してはCD並みの音声合成(16ビット2チャンネル，44kHzサンプリング）が可能です。ステレオ出力やマイク入力も実現されており，音声メイルのソフトウェアも付属しているようです。

画面出力には初めて「ディスプレイポストスクリプト」というものを採用しています。これはグラフィックディスプレイのための表示モデルであり，機種を意識せずにグラフィック関係のソフトを作成できること，ポストスクリプトをサポートしているレーザープリンタでそのままのイメージを出力できるという2つの利点があります。

ただ，直感的に表示が遅くなるだろうなと思うのですが，すでにそれを考慮したスピードアップのためのアルゴリズムやプログラムテクニックが開発されており，Nextでも採用されているそうです。実際，すごく速いそうですよ。

マルチメディアというパラダイムは，90年代にかけてますます重要なものになると思われるので，Nextはそれを先取りしたマシンといえるでしょう。

さて，このように入出力関係が充実し，CPUに高速なクロックの68030を搭載していても，CPU上のデータ転送に時間がかかっては意味がなくなってしまいます。原始的なアーキテクチャでは何をするにしてもメインCPUがデータを授受しているのですが，このNextではすべての入出力チャン

ネルにカスタムDMA（CPU抜きに直接メモリ転送を行う）チップを使用しています。これもリッチな構成といえましょう。

ソフトウェアでも多くの先進的なものを搭載しています。オペレーティングシステムに関しては，Machという，現在話題の分散型UNIXを採用しており，さらにその上にMacintoshを超えるようなユーザーフレンドリなインタフェイスを実現しているようです。

その統合的インタフェイスはNext Stepと呼ばれ，Window Server，Application Kit，Application Builder，Workspace Managerの4つから構成されます。これらはMacにおけるファインダ，ウィンドウシステム，ハイパーカードなどの要素を含むほか，よりオブジェクト指向の思想を強めたものであるようです。

実際，オブジェクト指向版のCであるObjective-Cを開発用言語として採用しています。これはプリプロセッサで，フィルタとしてCプログラムに変換してからGNU Cコンパイラにかけるわけです。超強力エディタとして有名なGNU EMACSも当然採用しています。

さらにアプリケーションとして辞書をまるごと，またシェークスピア全集までがオプションでなく提供されます。これこそ光磁気ディスクの御利益といえましょう。ペーパーメディアによる辞書や本の類がないという，いかにも未来生活を思わせるイメージがいきなり現実味を帯びてきます。

Nextについてちょっとだけそのハードとソフトを見てきましたが，さらに驚異的なのはご存じのようにその価格です。このようにぜいたくなスペックから見ると超低価格の6,500ドル（約78万円）という設定がなされました。キューブの中のキヤノン製光磁気ディスクの市場価格が6,000ドルということですから，いったいどうなっているのかと考えてしまいます。この価格設定こそ，「天才ジョブズ」の面目躍如といったところでしょう。

ちょっと心配なこと

さて，Nextに対する懸念を項目別に挙げてみましょう。

1）　製品のパフォーマンス

光磁気ディスクをシステムディスクとして使用した場合には，若干読み書きが遅いという感が否めないようです。これは，光磁気ディスクの本質に関わる問題かもしれませんが，インプリメントやバグなどの影響もNextのテストバージョンに表れていると見られます。まあ，少なくともハードディスクをキューブ中に搭載するという解決法はありますが，面白くありませんしね。

2）　モノクロディスプレイ

さしあたって問題として指摘されそうなところは，ディスプレイがモノクロであるということでしょうか。これはMacに対する批判で大きなウェイトを占めてきた点から見ても無視できないかもしれません。しかし，これについても一応すぐに開発するといってますし，実際来年中にもメドが立つでしょう。

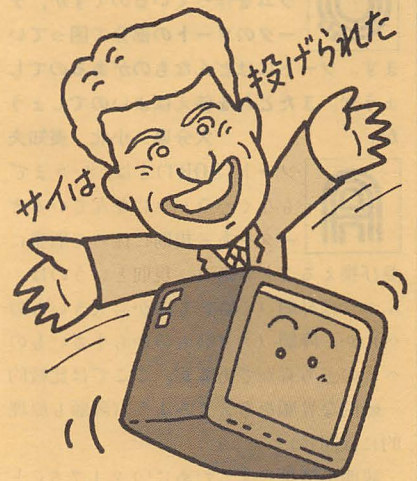
しかし，僕自身は以上のことについてなんとも思っていません（Macユーザーだなあ，しみじみと）。98などを使っても，やっぱりカラーだからというソフトはグラフィックソフト以外見かけませんし，あまり目にもよくないでしょうしね。90年代の半ば，あと8年もたてば，僕の意識も変わってくるでしょうか。

3）　Machを使う真の意味

現在の計算機の性能向上のための汎用の並列処理化に興味を持っている者としては，Machが使われていると知っていよいよNextへの好感を強めたものでした。マルチプロセッサ上での使用を想定した分散化したUNIXであるMach（あるいはその系統のUNIX）は，これから次第に大きな流れを作っていくのではないかという予感を持っているからです。

Machについては参考文献4などで紹介されています。新しい概念はスレッドと呼ばれるものです。これは，プロセスほど独立したものではなく，もっと結び付きのつよいようなものです。そしていくつかのスレッドが従来の意味でのプロセスを構成するといったモデルです。

ところでNextで採用されているバージョンはそのままではマルチプロセッサに対応できるものではありません，残念ながら。それではなぜMachをわざわざ採用したのでしょうか。

90年代のマシンなら必ずやマルチプロセッサをサポートするバージョンになると思

います。キューブ内に複数のCPUボードを搭載できるようになっていることですし。あるいはもう実験はなされているような気もします。

4）　使いたい（一番の心配なのだ）

実は，このマシンに関して一番心配なことは，いったいいつ僕がこのマシンを使える（自宅に所有する，大学で使う，人の家で使う，秋葉原で見る，などを含む）ようになるか，ということです。90年代のマシンを2000年（うむ，ちょうど40歳か）に入ってから触っても意味はありませんし。

Nextはさしあたり大学などの教育機関をターゲットとしています。だからこそこの価格を実現できたのでしょう。そういえばApple社は昔から大学に無料で配ったり，学生に半額で売ったりしていましたね。

でもまあ，スティーブン・ジョブズの夢は「個人個人にパーソナルなマシンを」ということだからいずれそうなるようにするだろう。そして，世界中の人や名古屋の人が欲しがり，もうちょっと値段が高くても売れるようになるだろう。といったところで，まあそのうち日本の人も触れるようになるでしょう。でもやっぱり心配だ，心配だ，心配だ。

参考文献

1）帰ってきたスーパースター，ニューズウィーク日本版，1988年10月27日号，8-13pp，TBSブリタニカ
2）BUG'S TOP，Bug News，1986年1月号，6-7pp，河出書房新社
3）Tom Thompson，"The Next Computer"，BYTE，Vol. 13，No.12，158-175pp，McGRAW-HILL
4）萩島茂直，UltramaxとMach，bit，1988年12月号，4-13pp，共立出版

QUESTION and ANSWER

X1turboで成績管理用のプログラムを作っているのですが，データのソートの部分で困っています。ソートにはどんなものがあるのでしょうか。またどれを使えばよいのでしょうか。

大分県　小北　美知夫

ソート(SORT)とは，いうまでもなくバラバラに並んでいるデータをある規則に従って順番に並び換えることである。規則というのは，ふつうは昇順（小さいものから大きいものへ）か，降順（大きいものから小さいものへ）のどちらかであるが，ここでは比較的一般的な昇順を考えてみよう（降順も原理的には同じ)。

説明を簡単にするために「9,2,1,7,5」というデータ列および一般的にN個のデータからなる列を考えてみよう。

1)　最小値選択法

まず誰でも考えつくのは最小値選択法(SELECTION SORT)というやつであろう。これは原理的にはいちばん簡単で，まず全体のデータの中からもっとも小さいデータを探し出し，それを先頭に持ってくる。続いて，最初の1個（もっとも小さいデータ）を除いた全データの中でいちばん小さいデータ（全体では2番目に小さいことになる）を探し，これを2番目に持ってくる。以下同様にして小さい順に頭から並べ，最後まで並び換えれば，はい終わり，というわけ。

これはアルゴリズム（というほどでもないが）は簡単だが比較および入れ替えを大量に行うため，実行速度はきわめて遅くなる。例のデータ列での動きは，

9,2,1,7,5
1,9,2,7,5
1,2,9,7,5
1,2,5,9,7
1,2,5,7,9

となる。

2)　交換法

次に一般的なのが交換法（バブルソート，BUBBLE SORT）というやつである。このバブルソートは隣りあうデータ同士をどんどん比較し，入れ替えることによってソートしてやろうというものである。

ソートしたいN個のデータからなるデータ列のうち，まず1番目と2番目を比べて，大きいほうを2番目に持ってくる。続いて2番目と3番目，3番目と4番目というふうにしていけば，結局最大のデータが最後（N番目）にくることになる。続けてまたまた1番目と2番目，2番目と3番目というふうにしてN－2番目とN－1番目を比較してやるとN－1番目に2番目に大きいデータがくることになる。このようにしていけば，最終的に小さいものから大きいものへと順序よく並ぶというわけである。

このアルゴリズムはちょっとわかりづらいかもしれない。小さいもの（あるいは大きいもの）をひとつずつ端から並べていくという点では選択法と同じであるが，違いは選択法では全データを調べてひとつ取り出して並べるのに対し，交換法では隣同士の交換により隣へ隣へと運んで並べるという違いがある。これは図1を見てみればわかるであろう。例のデータ列は，

9,2,1,7,5
2,1,7,5,9
1,2,5,7,9

となる。

3)　シェルソート

シェルソート（SHELL SORT）とは挿入法という方法の改良版にあたる。ちなみにシェルというのは人名である。挿入法というのはすでにある程度順序よく並んでいるデータ列にもうひとつデータを加えるときに，そのデータが入るべきところを探してその部分に挿入してやるといった方法である。説明は難しいが感覚的には理解しやすいはずである。まず初めに2番目のデータ。1番目のデータと比べて，2番目のデータのほうが小さければ1番目の前に挿入してやる。これで2つのデータからなる昇順列ができたことになる。あとはそれ以降のデータを入れるべきところに（先頭から順々に大小を調べて）挿入すればよいだけのことである。これを例で見てみると，

9,2,1,7,5
2,9,1,7,5
1,2,9,7,5
1,2,7,9,5
1,2,5,7,9

となる。実はなんのことはない，我々がトランプの手札を整理するときにやっているのと同じである。

以上のことからわかるように挿入法はデータがあらかじめ整理されていればされているほど効率がよい。この挿入法を改良したのがシェルソートである。シェルソートはソートすべきデータ列を何個かの部分列に分けてやることにより効率化を図ったものである。すなわち，部分列ごとに挿入法で整理してやり，全体をだいたい整理したあとで全体を挿入法で並べてやるのである。

で，部分列の作り方だが，N/2おきにデータを取り出して部分列を作る。次にはN/4おきに，そしてN/8……といけばやがてはひとつおき（全体の挿入法）になるというわけである。

要するに挿入法の2段構えという感じなわけだが，これを使えばまったく整理されていない数列でも比較的容易に整理できる。まあこの方法は文では説明しにくいのでかえってプログラムのほうがわかりやすいかもしれない。プログラムはリスト1である。

図1　選択法と交換法

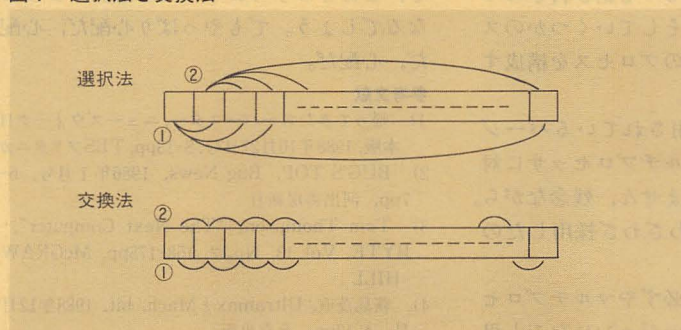
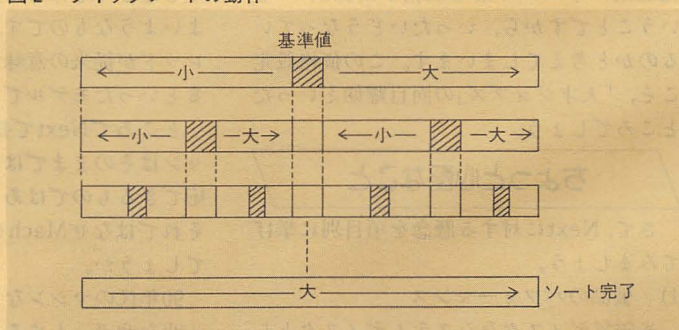



図2　クイックソートの動作

リスト1　シェルソート

```
1000 'shell-sort
1010 '
1020 no=50
1030 DIM dt(no)
1040 'data set
1050 FOR i=1 TO no
1060    dt(i)=INT(RND*100+1)
1070 NEXT
1080 'sort
1090 ds=no
1100 WHILE 1 < ds
1110    ds = ds ¥ 2
1120    FOR i=ds+1 TO no
1130       sdt=dt(i)
1140       s=i-INT(i¥ds)*ds
1150       IF s THEN lim=s ELSE lim=ds
1160       j=i-ds
1170       WHILE lim <= j
1180          IF dt(j) <= sdt THEN lim=j+ds
1190          j=j-ds
1200       WEND
1210       j=i-ds
1220       WHILE lim <=j
1230          dt(j+ds)=dt(j)
1240          j=j-ds
1250       WEND
1260       dt(lim)=sdt
1270    NEXT
1280 WEND
1290 'print
1300 FOR i=1 TO no
1310    PRINT i,dt(i)
1320 NEXT
1330 END
```

リスト2　クイックソート

```
1000 'quick-sort
1010 '
1020 no=50
1030 DIM dt(no)
1040 'data set
1050 FOR i=1 TO no
1060    dt(i)=INT(RND*100+1)
1070 NEXT
1080 'sort
1090 sp=1
1100 l(sp)=1
1110 r(sp)=no
1120 WHILE 0 < sp
1130    lp=l(sp)
1140    rp=r(sp)
1150    sp=sp-1
1160    WHILE lp <= rp
1170       li=lp
1180       ri=rp
1190       md=dt((lp+rp)¥2)
1200       WHILE li <= ri
1210          WHILE dt(li) < md
1220             li=li+1
1230          WEND
1240          WHILE md < dt(ri)
1250             ri=ri-1
1260          WEND
1270          IF li<=ri THEN SWAP dt(li),dt(ri):li=li+1:ri=ri-1
1280       WEND
1290       IF lp<=ri THEN sp=sp+1:l(sp)=lp:r(sp)=ri
1300       lp=li
1310    WEND
1320 WEND
1330 'print
1340 FOR i=1 TO no
1350    PRINT i,dt(i)
1360 NEXT
1370 END
```

また例を見てみると，

9, 2, 1, 7, 5
1, 2, 9, 7, 5
1, 2, 5, 7, 9

となる。

4)　クイックソート

今のところ（だいたい）いちばん速いといわれているのがこのクイックソートである。このクイックソートはアルゴリズムはプログラムにすると難しいわりに原理は簡単である。すなわち「データ列のうち，ある1個のデータ（基準データ）より小さい要素は左側に，大きいデータは右側にくるようにする」というものである。

具体的には基準は真ん中にとることが多い。データの先頭から基準値と比較し，基準値より大きければ右側に移してやる。データの終わりから基準値と比較し，基準値より小さければ左側に移してやる。こうして基準値の左に基準値より小さい列が，基準値の右に基準値より大きい列ができるわけである。そのそれぞれの列に対してまた新たに基準値を定め，ソートしてやればやがてはすべてが昇順に並ぶというわけである。これは図2を見てみるとわかりやすいであろう。これをプログラムにするとリスト2，また例のデータを処理してみると，

9, 2, 1, 7, 5
1, 2, 9, 7, 5
1, 2, 5, 7, 9

となるわけである。

以上主なソートについて述べてきたわけであるが，速さでいえば後ろのものほど速いということになる。が，逆に後ろのものほどプログラムは難しく複雑になる。よくソートは速さがすべてとばかりになんでもかんでもクイックソートを使う人がいるが，果たしてそれでよいのだろうか。確かにクイックソートは速い。しかし，たかが100個ぐらいのデータでは真価は発揮されない。データによってはほかのソートのほうが速い場合もあるのだ。

たとえば，すでに整理されている列に新しくデータを加える場合がいい例である。このような場合はシェルソートなどでもきわめて速い。しかもソートを使う場合はこのようなケースのほうがむしろ多いのではないだろうか。すなわち簡単なアルゴリズムでも，最初1回は遅くてもあとは（普段使うときには）速いということが可能なのだ。

要は必ずしも複雑なソートを使う必要はないということである。データの数，データの性質にあわせていちばんよいソート法を選んでやればよい。最後になったが，これらの説明はほんのさわりにすぎないことをつけ加えておく。この記述だけでは理解できないこともたくさんあろう。当然である。これだけで理解できるわけがない。プログラムを調べてみるなり，データの動きを図に書いてみるなり自分なりに理解に努めてほしい。また，ソートに関する本なども役に立つはずである。たとえば『ソーティング・ノート』(山本米雄著，日本ソフトバンク）などがある。今回のサンプルプログラムはここから引用している。

自分でいろいろとやってみて初めてソートがわかってくると思う。それだけソートというものは奥が深いものなのだから。

（華門真人）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせせることもありますので，電話番号も明記してくださいね。
宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

FILES Oh!X

このインデックスは，タイトル，注記――筆者名，誌名，月号，ページで構成されています。今月は各機種用の記事が多くて，一般の項がかなり縮まりました。相変わらずX68000の快気炎が目立ちます。

参考文献
I/O　工学社
ASCII　アスキー
テクノポリス　徳間書店
POPCOM　小学館
マイコン　電波新聞社
マイコン BASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶ ASCII EXPRESS シャープ，ISDN 対応静止画テレビ電話を開発
シャープの開発した ISDN 規格対応テレビ電話に関する速報。――編集部，ASCII，12月号，200p.

▶特集 CD-ROM が来る！
いよいよ一般的になってきた CD-ROM について，ソフトの進化には CD-ROM は不可欠だ，と特集を組んで解説。――編集部，LOGIN，11月4日号，120-127pp.

▶ The News File
SEGA の16ビットゲームマシン・メガドライブの紹介などパソコン関連のニュース。――編集部，LOGIN，11月4日号，156-163pp.

▶ The News File
夢の超スーパーウルトラパソコン Next の発表や，各メーカーのAXパソコンの紹介など。――編集部，LOGIN，11月18日号，220-227pp.

MZ-80K/C/1200/700/1500

MZ-80K/C/1200/700/1500
▶ SATURN
隕石に当たらないようにキーで操作して着陸するゲーム。――笹井進也，マイコン BASIC Magazine，12月号，138-139pp.

MZ-700/1500
▶コンバット
相手のハートを早くやっつけたほうが勝ち。コンピュータと対戦する陣取りゲーム。――小笹龍一，マイコン BASIC Magazine，12月号，140-141pp.

▶ある戦車の物語
敵をやっつけながら進み，要塞を破壊するゲーム。全5面。――カリット，マイコン BASIC Magazine，12月号，142-143pp.

MZ-1500
▶ CAR DRIVE
横スクロールする背景の中で車を操るカーアクションゲーム。――藤山健二，マイコン BASIC Magazine，12月号，144-145pp.

MZ-80B/2000/2500/2800

MZ-2000/2500
▶へりくん
3種類の爆弾を駆使して，敵戦艦をやっつけろ。――小門前伸司，マイコン BASIC Magazine，12月号，146-147pp.

▶さくらんぼ
さくらんぼが好物のノッブンの物語。フルーツの足し算をしながら面をクリアしていく。――石田学，マイコン BASIC Magazine，12月号，148-150pp.

MZ-2500
▶悪魔の赤い風船
悪魔につかまった天の使い，オンミツくんを脱出させるパズルゲーム。――謎のパズル大好きおじさん，POPCOM，12月号，244-250pp.

▶なんでも Q&A　シャープ MZ シリーズ編
MZ-2500でのプリンタコンフィギュレーションの設定の仕方について。――シャープ，マイコン，12月号，384p.

▶誌上公開質問状
MZ-2500用のグラフィックツールにはどんなものがあるか，など読者の質問に答える。――編集部，マイコン BASIC Magazine，12月号，70p.

▶ ICE BLOCK
氷の下に隠れたイチゴを10個取れば面をクリア。ペンゴじゃないよ。――電人，マイコン BASIC Magazine，12月号，151-152pp.

MZ-2861
▶なんでも Q&A　シャープ MZ シリーズ編
MZ-2861の up シリーズで使える文字フォントディスクについて。――シャープ，マイコン，12月号，384p.

▶なんでも Q&A　シャープ MZ シリーズ編
MZ-2861で，チャート up にビジレス AD で作成したデータを読ませる方法について。――シャープ，マイコン，12月号，384-385pp.

▶なんでも Q&A　シャープ MZ シリーズ編
MZ-2861のエミュレーションソフトのバージョンアップの方法について。――シャープ，マイコン，12月号，385p.

▶なんでも Q&A　シャープ MZ シリーズ編
MZ-2861のアプリケーションソフトのデータの形式について。――シャープ，マイコン，12月号，385p.

X1/X1turbo/Z

X1シリーズ
▶西安
リアルタイム2プレイ上海風ゲーム。――POP・ART，テクノポリス，12月号，93-96pp.

▶新製品レポート　シャープ X1turboZ III
X1turboZ シリーズの新製品，X1turboZ III (CZ-888C) の仕様，価格などについて紹介。――編集部，マイコン BASIC Magazine，12月号，46p.

▶誌上公開質問状
X1F/20のマニュアルについて，また X1G にマウスを接

世の中は一般人の目に触れないところで刻々と進もうと頑張っている。特に「情報」がらみの世界ではそうだ。本書もそうした世界から提出されたものの一種だが，しかつめらしい表題と中公新書というブランドから，なにやら専門書の匂いをかきわけると，それは幅広く著者の思うところを網羅した散文のようであり，技術よりも概念を伝えようとしていることに気づく。工学者特有の人に対する甘さが気になるにしても，先端の情報処理環境を知り，そのトップクラスの人が次になにをしようと考えているかわかるだけでも読む価値はあろう。本質的に難解な話ではなく深い突っ込みもないので，頻出する専門用語に対する注と，もっと平易な具体例さえあれば，より多くの人が面白く読めたのではないかと思われる。

やはり鍵となるのは「ネットワーク」，（広義のファジィやニューロンを含む）「人工知能」，それからデータ量の差の克服（文字やグラフィックや音声など異メディアを同等に扱う！）の3つだった。　（K）

戦略的創造のための情報科学
坂井利之著　中央公論社刊
新書判　348ページ　560円
☎03(563)1261

続するにはどうするか，などの質問に答える。──編集部，マイコン BASIC Magazine，12月号，71p.

▶酔っぱらいブギ

主人公は酔っ払い。婦警さんにつかまらないように画面の酒を全部飲むパズルゲーム。──T.AKUMA，マイコンBASIC Magazine，12月号，192-194pp.

▶ SLIP OTHELLO

オセロを題材にしたパズルゲーム。──崎山高博，マイコン BASIC Magazine，12月号，195-197pp.

▶ドラゴンスピリット　エンディング

ゲームミュージックプログラム。──上田順一，マイコン BASIC Magazine，12月号，214-217pp.

▶最新ゲーム徹底解剖!!

サイオブレードなどの最新ゲームを詳しく紹介。──編集部，LOGIN，11月4日号，104-119pp.

▶最新ゲーム徹底解剖!!

戦国ソーサリアンの開発者インタビューから，章単位での攻略法まで。──編集部，LOGIN，11月18日号，150-153pp.

X1turboシリーズ

▶ X1turbo にハードディスクを！

X1turbo シリーズでハードディスクを使うための，インタフェイスボードの工作記事と，ハードディスク対応にするためのソフトの改造法について。──今雪寛，I/O，12月号，216-223pp.

▶なんでも Q&A　X1/X1turbo/X68000シリーズ編

turbo CP/M の起動時にシステム辞書を使えるようにするためのシステムの書き換えについて。──シャープ，マイコン，12月号，383p.

X68000

▶描画プログラムジェネレータ

マウスで描いた図形を XC 用のソースプログラムに変換するプログラム。XC 用のソースリストを掲載。──X1 Player's Club ずん，I/O，12月号，123-137pp.

▶ 1 枚のプリンタ用紙に 4 倍のリストを出力する「圧縮リスタ」

X68000とプリンタ VP-800PC を使った，リストの圧縮文字出力ソフト。C コンパイラ用のソースリストとして掲載。──来夢雷人，I/O，12月号，154-159pp.

▶環境ソフト　居座り時計

アセンブラを使った環境ソフト，居座り時計。システム常駐形プログラムの典型として，ソースリスト付きの掲載。──石川一彦，I/O，12月号，197-199pp.

▶コンピュータウイルス対策とワクチン

X68000にウイルスが出て来たときの対策に，コンピュータウイルスとはいかなるものかの説明と，実際に無害なウイルスとそのワクチンの作成を行っている。──市原昌文，I/O，12月号，244-245pp.

▶ X68K　Information　Shop

シャープから発売された純正 MIDI ボード，CZ-6BM1とその対応ソフト Musicstudio PRO-68K について，主なスペックの解説，それに計測技研の開発したMelody Box付属ソフトの紹介など。──編集部，ASCII，12月号，317-318pp.

▶ X68K　Report　Shop

ジェーイーエルの発売した純正アセンブラ対応プリプロセッサ PP68K と，シティソフトから発売のアセンブラ CMA68K を開発環境の向上という視点から解説。──編集部，ASCII，12月号，320-322pp.

▶ X68K　Programmer's Shop

OS-9/X68000入門編として，OS-9/X68000と X68000の標準 OS，Human68k との違いについての解説。──編集部，ASCII，12月号，323-324pp.

▶ソフトウェアライブラリ

X68000用の課題に応募してきたプログラムの紹介。今月はパターンエディタ PE，スプライトエディタ SPED.X，キャラクタエディタ ED1/2の 3 本。──編集部，ASCII，12月号，325-332pp.

▶ X68000ワールド

最新ソフトの紹介。ザ・キング・オブ・シカゴ，沙羅曼蛇，サンダーフォースII，Sampling PRO-68K，道化師殺人事件。──編集部，POPCOM，12月号，81-87pp.

▶グラフィックツール実力診断

C-TRACE68の紹介がされており，レイトレーシングのアルゴリズムから，C-TRACEのデータの作り方まで細かく解説されている。──紀要介，マイコン，12月号，148-153pp.

▶ X68000マシン語入門

今月は，2 進化10進数を扱うための BCD 命令と IOCS を使ったグラフィック命令の使い方（前編）についての解説。──高橋雄一，マイコン，12月号，181-189pp.

▶なんでも Q&A　X1/X1turbo/X68000シリーズ編

Sampling PRO-68K で取り込んだサンプリングデータを X-BASIC の BEEP 音として使う方法について。──シャープ，マイコン，12月号，382p.

▶なんでも Q&A　X1/X1turbo/X68000シリーズ編

X68000用の NEW Print Shop PRO-68K でできることについて。──シャープ，マイコン，12月号，382-383pp.

▶なんでも Q&A　X1/X1turbo/X68000シリーズ編

X68000でコンピュータ画面をビデオ録画するために必要なオプションについて。──シャープ，マイコン，12月号，383p.

▶風の聖塔

五郎太を出口へ誘い出すキャラに凝ったゲーム。──軽澤誉，マイコン BASIC Magazine，12月号，198-200pp.

▶グラディウスIIステージ 8

ゲームミュージックプログラム。──粟田英樹，マイコン BASIC Magazine，12月号，207-209pp.

▶チャレンジ！　X68000

沙羅曼蛇，サンダーフォースII，たんばを紹介している。──川野俊充，マイコン BASIC Magazine，12月号，296-297pp.

▶ SOFTWARE REVIEW

A 列車で行こうIIのマニアックな実戦的解説。──都築“シベリア”てつや/足軽くんサナダ，LOGIN，11月4日号，34-35pp.

▶ X68000新聞ときめき関西特集

関西系ソフトハウスの特集，よみものドラスピ日記，デスプリンガー，MIDIシステムMelody Box，沙羅曼蛇の紹介など。──編集部，LOGIN，11月4日号，146-151pp.

▶ NEW SOFT

今夜も朝まで POWERFUL まあじゃん X68000版の開発中の話題。──編集部，LOGIN，11月18日号，14-15pp.

▶ SOFTWARE REVIEW

オリンピックは終わっても相変わらず盛んな熱血高校ドッジボール部について。──モデラー松本，LOGIN，11月18日号，32-33pp.

▶ X68000新聞

今夜も朝まで POWERFUL まあじゃんとウォーニングの紹介や，熱血高校ドッジボール部と沙羅曼蛇の秘技，C-TRACE の初心者向け講座など。──編集部，LOGIN，11月18日号，208-213pp.

ポケコン

PC-G801

▶誌上公開質問状

PC-G801でオリジナルキャラを表示するやり方などの質問について。──編集部，マイコン BASIC Magazine，12月号，71p.

PC-1245/1251/1255

▶ FISCO

バイクで富士スピードウェイを 3 周するゲーム。──今村晃，マイコン BASIC Magazine，12月号，203p.

PC-1360K

▶宛名書きプログラム

11月号の住所録プログラムで入力したデータをポケットディスクに保存し，プリンタ CE-515P で宛名を印字するためのプログラム。──塚田洋一，マイコン，12月号，348-352pp.

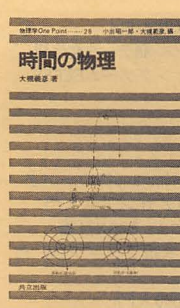
PC-E500

▶失われた王様の杖

オール BASIC のロールプレイングゲーム。猟人よ，ドラゴンを倒せ。──うんとと 3 号，I/O，12月号，204-207pp.

ニワトリの歯（上・下）

現在，米国で最も人気のある科学者のひとりS.J.グールドによる第 3 エッセイ集。ニワトリに歯がはえる可能性はあるのか。シマウマの縞はどうやってできるのか。生物の絶滅はどうして起きたのか。収録された30篇のエッセイは，いずれも読者を知らず知らずのうちに生物学の核心に引きずり込む。進化は科学が生んだ数少ない偉大なアイデアのひとつだという著者の言葉が実感される。

S.J.グールド著　渡辺政隆・三中信宏訳
早川書房刊　A5判　上304ページ　下312ページ
各1,500円　☎03(252)3111

時間の物理

物理といってもさほど難しい数式は出てこない。もちろん本書の主目的は時間の物理的考察だが，はしがきに，恐山のイタコが呼び寄せる死者にはもう時間というものはないのだ，というエピソードを紹介しているところからもわかるように，時間とはいったい何だろうかという疑問を，まず人間の初歩的・感覚的なものとしてとらえている。なぜ時間は一方向にのみ流れるのか。素直に不思議に思う人は読んでみるといい。

大槻義彦著　共立出版刊
B6判　144ページ　980円　☎03(947)2511

BACK ISSUES バックナンバー案内

ここには1988年1月号から1988年12月号までをご紹介しました。現在，1987年4, 8, 10, 11, 1988年1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12までの在庫がございます。バックナンバーおよび定期講読のお申し込み方法については，本文174ページを参照してください。

1988

1月号
特集　MZ&X拡張ボードの活用
すべての道はI/Oに通じる/MZでX1用ボードを使う
1987年度GAME OF THE YEARノミネート発表
●MZ-2500用 ALGO SPACE BLUSTER SG
●LIVE in '88 ドラゴンスピリット/悲しきチェイサー
BASICリレー連載　半熟FORTRANはいかが
X68000BASIC入門　グラフィック炎上
マシン語体操1・2・3　データ構造を考えよう
全機種共通システム　FuzzyBASICコンパイラ奥村版

2月号
特集　グラフィック画像の冒険
X1/turboCGアニメ/トリフォニーで立体モデル
X68000グラフィックデータ/QUICK MZ PAINT他
X68000あなたの知らない世界　辞書構造/WORD POWER
マシン語体操1・2・3 Lispインタプリタ(1)
●NEW Z-BASIC詳報　その名はZ-BASIC
●LIVE in '88 グラディウス2
●SHORT ACCESS THRILLING/POMカードポーカー
全機種共通システム　シューティングゲームELFES

3月号
特集　コンピュータサウンド“楽”入門
X1/turbo MIDIインタフェイスの製作
MZ-2500 Super Keyboard/VIPサウンドデータ公開
Oh!X LIVE SPECIAL 組曲「Ys」/Raspberry Dream他
THE SOFTOUCH Might and Magic/Hyper UD
オブジェクト指向のゲームプログラミング
X68000BASIC入門 奇襲アニメ作戦
X68000あなたの知らない世界 未公開IOCSの解析
全機種共通システム 構造型コンパイラ言語SLANG

4月号
特集　不思議の国のゲーム学
決定！ 1987年度GAME OF THE YEAR
ピコピコゲーム春場所/GAME REVIEW 10本他
新製品　X68000ACE-HD/カラースキャナCZ-8NS1
X68000あなたの知らない世界　microEMACSの移植
●MZ-700 SPACE BLUSTER FX
●LIVE in '88 Moonlight Serenade/Long Night 他
全機種共通システム　デバッギングツールTRADE
シミュレーションウォーゲームWALRUS

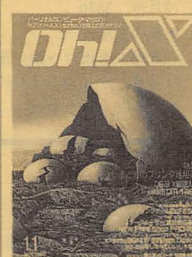
5月号
特集　BASIC入門「再検証」
BASICの歴史と意義/栄光のHuBASIC
黄金のBASIC入門プログラム/プログラミング用語集
ミュージックプログラマへの道/レイトレーシング
特別企画　言わせてくれなくちゃだワ
●新製品　X68000ACE/ACE-HD
●LIVE in '88 GET WILD/BOOM BOOM/SDI
●SHORT ACCESS 3Dボクシング/マシン語データ文生成
全機種共通システム　シューティングゲームELFES

6月号　創刊6周年記念
特集　システム環境を考える
8ビットパソコンの開発環境/Human68kのシステム環境/システムを読むためのアセンブラ入門
特別企画　究極の8ビットパソコン 8RON計画
THE SOFTOUCH X68000用日本語ワープロEW 他
●付録「あぶない福袋」
マシン語体操1・2・3 番外編　Lisp80入門
X68000BASIC入門　捨て身のミュージック
全機種共通システム　構造化言語SLANG入門 他

7月号
特集　実践C言語からの誘惑
入門C言語/実録Cプログラミング/XBAS to C
THE SOFTOUCH ソーサリアン/ゼリアード/アルギースの翼/SUPER大戦略/3大麻雀ソフト 他
●Oh!X LIVE in '88/SHORT ACCESS
新連載 C調言語講座PRO-68K まずはprintfより始めよ
あなたの知らない世界 OS-9/X68000/Sampling PRO-68K
全機種共通システム　構造化言語SLANG 入門(2)
マルチウィンドウドライバMW-I

8月号
特集1 真夏の夜の数値演算
コンピュータの数値表現/応用グラフィック歪められた光/AD PCM音の数学/数値演算プロセッサ用ドライバ 他
特集2 MIDIサウンドプログラミング
MIDIの基礎とボードの製作/MIDI対応シーケンサ
THE SOFTOUCH　新連載　われら電脳遊戯民　他
猫とコンピュータ第26回　ボクはかぐや姫？
新連載　Z80マシン語ゲーム工房
全機種共通システムマルチウィンドウエディタWINER

9月号
特集　半期に一度のグラフィックバザール
CGアニメの手法入門/ワイヤフレームによる3D/X68000スプライト/画像処理の基礎知識/turbo RAY TRACER/MZ-2500用グラフィックエディタDMACS
THE SOFTOUCH C-TRACE68/SAMPLING PRO-68K他
C調言語講座PRO-68K(3)　謎の低次元グラフィック
MIDI活用テクニック(2)　割り込みによるMIDI通信
Z80マシン語ゲーム工房(2)　応用への基礎固め
全機種共通システム　ラインエディタTED-750/WINERの拡張

10月号
特集　百花繚乱ゲームバトルロイヤル
最新ゲーム総登場　ハイドライド3/A列車で行こうII/たんば/熱血高校ドッジボール部/フルスロットル他
MZ-700用SPACE HARRIER
●Oh!X LIVE 1974(16光年の訪問者)/瑠璃色の地球/二人のゼネレーション/バッハのアリア
MIDI活用テクニック(3)複数の音源を操るテクニック
C調言語講座PRO-68K(4)/Z80マシン語ゲーム工房(3)
全機種共通システム　SLANG用拡張ライブラリ/MANKAI

11月号
特集　いまどきのプリンタ活用術
メカニズムを理解しよう/制御コード/文字と図形の混在印字/拡大文字のスムージング/外字登録ツール/S-H COPY/グラフィックのモノクロ出力/X68000のCOPYキー/オリジナル印刷キット/試用レポート
THE SOFTOUCH　NEW Print Shop PRO-68K 他
OS-9/X68000入門(1)　OS-9ってなに？
●STAR TREK for X68000
全機種共通システム　シューティングゲームELFES IV

12月号
特集　パソコンはいま音楽の領域へ
なぜ自動作曲か/心地よい雑音の話/和音の読み方/美しい響きの要素/4分音符は歌い始める/古くて新しい音楽形式/FM音源の仕組み/Melody Box/MusicBASIC
●さよなら Live in '88 バッハ イタリア組曲ほか6本
●Oh!X 1周年記念特別企画「ちょっとあぶない福袋」
OS-9/X68000入門(2) OS-9のオペレーション環境
Z80マシン語ゲーム工房/C調言語講座PRO-68K
全機種共通システム　ソースジェネレータ SOURCERY

FROM READERS TO THE EDITOR

あっという間にOh！Xはもう新年号。これからもまた皆さんと楽しい1年を過ごしていきたいと思っています。この冬は例年になく寒い冬となりそうだけど，STUDIO Xはいつでもホットで明るい話題を提供し続けるページでありたいですね。

◆11月号の「S-HCOPYルーチン」は有難く使わさせていただいています。ここでちょっと136桁プリンタをお持ちの方は試してみてください。まず63ページのリスト4にある，キングサイズの追加訂正プログラムを次のように変更してください。最初に680行の後ろに付いている「POKE&HE 13B,4」を削除。そして785行をカット。6080行を「DATA 255, 8, 3, 67, 4, 640〜」と変更します。そうしてRUNすると，キングサイズのハードコピーが1枚の紙に一発でとれます。これはなかなか爽快ですよ。

米沢 賢一 (19) 宮城県

特集で発表してすぐに，こんなお便りをいただけてとても喜んでいます。このほかにも，ちょっと手を入れただけで活用できるような改造法があれば，いつでもお送りくださいね。

◆我が家にプリンタが来る。親父に「こーてくれー」といったら，1週間くらいして突然にエプソンのVPのパンフレットを見せて，「2，3日で来るぞ」というのです。モノをねだると普段は発狂する親父がどうしたというんだ。よほど機嫌がよかったか，血迷ったかのどちらかだと思います。でもエプソンのプリンタケーブルってうちのturboにつながるんだろうか。それに136桁では置き場もない。でも，おかげで来月からこのババッチイ字を世間にさらさなくてもすみそうです。

藤田 慶 (20) 兵庫県

これはどうも藤田君のお父さんって，Oh！Xをこっそりと読んでいたのでは？ だから11月号を読んで，プリンタを買ってくれたんじゃないのかな。そうするとだよ，今月あたりは突然メガドライブが家に……，ってなわけにはやっぱりいかないだろうね。

◆「われら電脳遊戯民」の倉持さん，ファミコンの話題がタブー視されているパソコン雑誌において，よくぞあそこまでほめてくれました。私もドラクエは3作とも解いたのですが，なかでもIIIは非常によかった。Iをやっていないと，終わりの感動は少ないけど，確かに駆け落ちにはぶったまげたし，幽霊船というアイデアは素晴らしかったし，不死鳥の背中に乗って大空を飛んだときは気持ちよかった。そして夜になって危ないお姉さんにパフパフしてもらえるのもいい。そしていま私はファミコンウォーズにのめり込んでいる。ファミコンでなにより嬉しいのは，未だにカセットを見捨てないことである。

横田 紀明 (21) 山口県

カセットを見捨てなくても，ディスクを見捨てるんじゃないかという説も……。それはともかくファミコンだろうが，メガドライブだろうが，みんなで楽しくやればいいんじゃないのかな。でも，スーパーファミコンやら，スーパーPCエンジンなんていうのが出てきたらどうしましょうかね？

◆X68000用の「STAR TREK」には笑ってしまった。リストを追っかけながら，メッセージを読んでいくとついつい吹き出してしまう。これは近年まれに見るソフトだと思います。

峠 保 (22) 東京都

まさにリストを読んでも笑えるソフトなのかもしれませんね，このSTAR TREKって。

◆「ELFES IV」は，前作のIやIIと比べてとても完成度が高いと思う。最初は敵の弾が見えなくて苦労していたが，回を重ねるごとに次第に見切れるようになり，1面は楽勝でクリアできるようになった。と，思いきや2面のあのイモ虫の大群に悩まされている今日このごろである。ELFESの作者の青木さんは，いつもこんな難しいゲームばかり作っているが，ひょっとしてプレイヤーとしてはプロ並みの腕を持っているのではないのだろうか。などと考えながらイモ虫の大群を相手にしている私なのであった。次回のELFES IIIはもっと読者思いのゲームにしてください。

佐藤 敏晴 (18) 秋田県

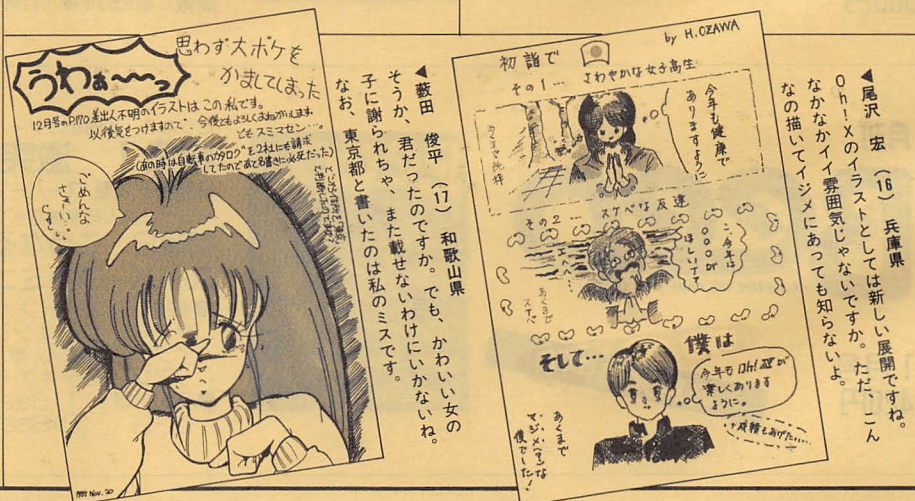
この佐藤君のセリフを，そっくりそのままテクノソフトさんにもいってほしいような気がします。サンダーフォースIIの5面の横は，とても人間技ではクリアできそうにありません。

◆11月号のプリンタ特集はなかなかタメになった。1年前にturboを買って，ずっと2000文字対応ディスプレイを使っていたのだが，turboZ'sSTAFFを400ライン専用だと知らずに買ってしまったのをきっかけに，夏休みに遊ぶのをガマンして，測量のバイトで4050文字対応のディスプレイテレビ，マウス，FM音源，カラーイメージボード，XE1-PROなどを次々と買った。おかげでturboZ'sSTAFFも思う存分使えるようになった。が，なにかもの足りないような気がした。そう，プリンタを買い忘れていたのだ。今月の小遣いは生ディスク代に消えてしまったので，この調子だと正月に郵便配達のバイトでもしなければならない。そこでお願いがあります。プリンタ特集をやったときには，プリンタをプレゼントするべきです。というわけで，また来月もプリンタ特集をやってください。

桜井 隆志 (17) 北海道

◆私はいま，漢字のフォントを新しく作っています。自分が必要な文字だけを作っているのでまだ300文字くらいしかありませんが，これからもっと増えることでしょう。16ドットのROMフォントを拡大して，ちょっと手を加えてディスクにセーブするだけなので，あまり手間はかかりません。なんて私はその筋なのでしょう。だから，その筋キーホルダーをください。

◀藪田 俊平 (17) 和歌山県
そうか，君だったのですか。でも，かわいい女の子に謝られちゃ，また載せないわけにいかないね。なお，東京都と書いたのは私のミスです。

◀尾沢 宏 (16) 兵庫県
Oh！Xのイラストとしては新しい展開ですか。ただ，こんなの描いてイジメにあっても知らないよ。

岩瀬　英朗　(18)　神奈川県

本当に，先の桜井君にしろ岩瀬君にしろ，最近の若い者は……。いかん，いかん，オジンくさいセリフを思わず並べてしまいそうになってしまった。

◆セガ・メガドライブは凄い。68000とＺ80を積んでいるのが凄い。値段が安いのが凄い。とにかく凄い。これはスーパーファミコンへの挑戦状だ。過去のしがらみに捕らわれたスーパーファミコンなんてくそくらえ。がんばれメガドライブ！　私は常に挑戦者の味方なのだ。このあと引き続き，Ｘ68000とともに80386を積んでスピードアップしただけの98RAと戦っていく所存であります。　西尾　郁彦　(20)　愛知県

◆いまさらながら「ノルウェイの森」を読みました。恋愛小説だということでしたが，私はこの本を読んで死というものについて考えさせられました。なぜ，彼らは自殺したのか？　そのことが妙に印象深く心に残っています。実は彼らのほうが正常で，この異常な世の中についていけなかったのでしょうか。私にはいまの世の中が少し狂っているような気もします。具体的になにが，とはいえませんが……。皆さんもそう思いませんか？　鈴木　則夫　(18)　埼玉県

うーん，鈴木君のいうように，あの本は村上春樹の代表作となりそうですね。でも，このままいくと昔の遠藤周作と同じようなパターンにもなりかねないので，彼の作品がこの先どうなっていくかのほうが楽しみともいえそうです。

◆フィッシャーマンズスープレックス，ジャーマンスープレックス，パワーボム，ノーザンライト式体固め，ヒザ十字固め，アキレス腱固め，スモールパッケージ，タイガースープレックス，延髄斬り，卍固め，ジャパニーズレッグロールクラッチホールド……，これらはすべて先日行われた，私ことテロリスト西谷vsケンタウルス奈良岡の無制限１本勝負のなかで飛び出した技の一部です。結果は21分30秒，ラリアット２連発からコブラクローでフォールして首固めで私の勝ち。試合後，ケンタウルス奈良岡はラッシャー木村調で「やい西谷，俺が今日負けたのは，借金で首が回らないところを首固めなんかかけやがるからだっ！」と，のたもうておりました。これまでの彼との対戦成績は１勝１敗１分となっています。ご希望であれば，過去の対戦中継をいつでもお送りします。できれば次回は，我々のレフリーをやってくれているクィーン関口の私生活についてお届けしようと思っています。

西谷　久範　(20)　宮城県

あまりこれまでの対戦中継は希望したくはないのですが，プロレスファンの清水和人さんは喜ぶかもしれないので，ひとまず送ってみてください。それにしても21分もの勝負とは，よくやりますね。

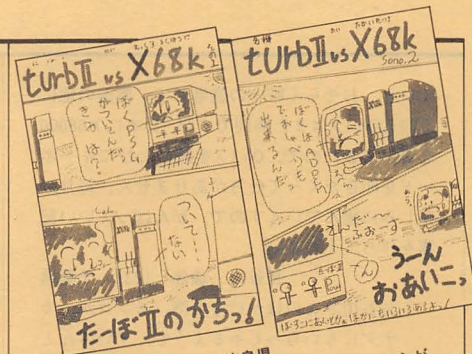
◆パリーグは西武が優勝してしまいました。しかし，今年は本当に近鉄に優勝してほしかった。特に近鉄ファンだというわけではないこの僕を，あの最終戦ではテレビの前に釘付けにしてくれたのだ。最後のダブルプレーのときには思わずタメ息をついてしまいました。でも，選手の皆さん本当にいい試合をありがとう。

国政　寛　(17)　大阪府

ほんと，最終回に守備についた近鉄の選手たちを見ていると，思わず涙が出てしまいそうになりました。これからは，試合によっては引き分けをなくしてしまえばいいと思うんですけどね。

◆王さんが辞め，阿南さんが辞め，高田さんも辞め，掛布さんも引退し，山田さんも引退表明をし，南海は福岡に行き，阪急はオリエント・リースに身売りされ，近鉄は優勝できず，阪神は連続最下位……。挙句の果てに，僕は女の子にフラれるし，ああ，世の中はいったいどうなっているんだっ!!　以上，受験生のひとり言でした。　福島　義浩　(19)　滋賀県

◆源平の超ウラ技をお教えしましょう。なんと影清がマイケル・ジャクソンなみにムーンウォークするのです。やり方は簡単。Ａボタンを押して後ろを向いて，そのままＡボタンを押しながら前へ進むのです。どうです，コリャ凄い。あー，なさけなや。吉本　浩二　(19)　広島県

まだ，やってみてはいないけど，これは結構笑えそう。

▲寺島　昭栄　(15)　徳島県
これは勝負あった。いやー，ほのぼのとして心が洗われるようですね。無邪気なturboIIくんにかかっては X68000もかないません。

◆11月号145ページの下村さんへ。デゼニランドは，ロッカーの横にあるカンをMOVEすると鍵が出てきますから，鍵に書いてある番号のロッカーを開けてみよう。おっ，お金が出てきた。そしたら，そのお金でダフ屋からチケットを買えばそれでOKだ。

榊　龍太郎　(17)　神奈川県

◆「神戸商科大学」。皆さんのなかでこの大学の名前を知っている人はいったい何人いるのでしょうか。「某三流私大」と思われがちだが，しかし，本当のところは一流半の，兵庫県立の公立大学なのである。ぜひOh！Ｘの力で有名にしてやってください。　長田　大輔　(18)　兵庫県

◆おおっ！　うちの大学の名前がOh！Ｘに出ている。まさか知能機械概論に，豊橋技術科学大学の名前が載るとは思っていなかった。ところで，そこに書いてあった例の匂いですが，あれはご近所に住むコケコッコの匂いだそうです。面白いことにあれが匂うと，だいたい次の日は雨なんですよね。実に情けない天気予報です。

小野　義之　(20)　愛知県

Oh！Ｘと受験生って，切っても切れない関係にあるわけだから，このようなお話は大学選びのいい参考になるかもね。

◆今日，前期の成績発表がありました。しっかりと４単位落としてしまいました。進級には関係ない単位ですけどね。ちなみに落としたのは藻類生態学と水質学。さて，私の通っている学校は？　　大田　敏郎　(21)　東京都

うーん，単位を落とした言い訳をクイズにしてごまかそうとしているな。ちなみにこのクイズに正解の方には，大田君から４単位が贈られるそーです（んなわけないって）。

◆Oh！Ｘ LIVEに「NHKみんなの歌」でも歌っている，種ともこをリクエストします。

野田　敏之　(17)　神奈川県

これこれ，あのコーナーは曲のリクエスト受け付けるためにあるんじゃないんだから，自分でプログラムを作って投稿することも少しは考えなさいってば。

◆Oh！Ｘって，ペットショップになったり，結婚相談所になったり……，うーん，ようわからん。

石田　貴志　(17)　大阪府

◆東ヨーロッパはブルガリアにてOh！Ｘを読ん

いつだって全速力 シェイク

イラストでお馴染みの山崎潤一君所属の「パソコンクラブWATER」から，会報「シェイクVol.4」が到着したのでご紹介します。ＣＧで描かれた表紙のなかに収められた情報は，それぞれパソコン講座，アニメファンのためのコンピュータ傾向と対策，IOCS活用法，Ｘ68Ｋ顚末記などなど，役立つものや笑わせてくれるものなどが工夫を凝らしてまとめられていて，なかなかに楽しく読ませてくれます（あの田村憲生の投稿もあるぞ）。

このように，最近のサークル活動の内容をご紹介する意味でも，これからOh!X編集室に届けられた会報は順次ご紹介していきたいと思っていますので，活動中のサークル方は遠慮なくドシドシお送りください。お待ちしています。

でいる，おそらく唯一の読者です。異国の地にて読むOh！Xはまた格別ですが，日本にＸ68000を置いてきたので，Oh！Xを読むたびにキーボードを叩きたくって仕方がありません。毎月，楽しみに読んでいますので，がんばっていい記事を作ってください。

長谷川　諭伴　(23)　千葉県

長谷川さんのほかにも台湾，モンゴル，カナダにもいらっしゃるようで，ずいぶんと国際的に読まれていて嬉しい限りです。でも，ブルガリアのパソコン事情っていったいどうなっているのでしょうね。今度また機会があったら教えてください。

◆11月号のこのコーナーに載っけてもらった田中です。結局，国体に行ってきました。試合の結果は予想どおり5，6位以下だったのですが，とても楽しい1週間でした。いろいろとよその学校の選手と友だちになって，いまでは電話や手紙で連絡を取り合っています。冬休みには会う約束もしています。やはりOh！Xのメッセージどおりにしてよかった。

田中　哲也　(18)　兵庫県

私も経験あるけど，学校が違っても同じ競技なんかに出場している仲間とはすぐに親しくなれて，勝っても負けても楽しいものなのです。

◆いま「タイムエスケープ」という本を読んでいます。破滅が迫った地球を救うため，光より速い粒子を使って過去の世界と通信しようという内容です。まだ途中までしか読んでいませんが，タイムパラドックスを回避する理論を理解しようとすると頭が痛くなってしまいます。でも，最後にあっといわせてくれそうで楽しみ。いままでにも，タイムパラドックスを扱ったアドベンチャーゲームはいくつかありましたが，この本みたいなリアルな雰囲気を持ったストーリーのゲームを遊んでみたいと思います。

尾崎　誠　(20)　福井県

◆最近，聞いた話によると，富士通の大反撃が32ビットで始まりそう。思えばひと昔前，FM-7というやつが私たちの憧れのマトでした。いまでは若くしてＸ68000を購入できる人たちが多いみたいで，うらやましい限りです。

後藤　仁志　(34)　岐阜県

最近，パソコン，ゲームマシンともにさまざまな噂が飛び交っているようで，年明けあたりはずいぶんと賑やかになりそうな予感がします。今月のメガドライブのタイトルじゃないけど，Ｘ68000のライバルっていったい誰になるんでしょうか。

◆「たーん，たーん，たーよし，いきよしたーよーし，おーなーじのむならやっぱりたーよーし」というCM知っていますか？　大阪近郊の人にしかわからないとは思いますが，セピア色に変色した画面といい，異常に若い大久保れい(名前の漢字忘れた)といい，強烈なインパクトがあり，歴史に残る名CMだと思います。

本田　泰啓　(18)　大阪府

なんか，歌詞を聞いただけでも凄いCMみたいですね。ところで大久保れいって，いったい誰なんですか？

◆素朴な質問をここで3つ。

1）ブラスティーやα（アルファ）で有名なスクウェアは，いったどこに行ってしまったんでしょうか。

2）日本テレネットのアルバトロスに付いていた会員証を使った企画は実在したのでしょうか。

3）最近，誌上では「～ぢゃないか」とか，「～なのぢゃ」などという表現が流行っているようですが，「究極超人あ～る」なんて知るわけがない年輩の読者の方から「言葉遣いがなっとらん」と，お叱りの言葉をいただいたことはありませんか？　以上3つについてお答えください。答えてくれないなら，プレゼントちょーだい（なんてあつかましいヤツ）。

飯村　幸男　(18)　京都府

あつかましいのを多少は自覚しているようだから，質問には答えてあげよう。1）どこにも行ってません。2)知りません。3）ありません。ちなみにあれのOh！X内での流行もとは，スピリッツの「YAWARA」です。

◆76坪の土地に35坪の家を建て，新車を1台，中古車を1台買い，部屋には27インチモニタにVHS，β，オーディオ……。妻1人，子供3人を抱え，果たして冬のボーナスでＸ68000に買い換えることができるだろうか!?

◀大野　真実　(?)　静岡県
出た～っ、やっぱし次は「たんばじゃないかと思っていたんですよ。それでは皆さん、ごいっしょに、ハイッ「たんばぁふぉ～す!」

海野　伸司　(28)　群馬県

好きにしてください（今月はこのパターンばっか)。

◆できれば質問箱で取り上げてください。Oh！Xを買うと，まずSTUDIO Xに自分の名前を探してしまう私は異常でしょうか？　（タコ足男・札幌市)。うーん，無謀だ。

畦地　真太郎　(18)　北海道

まっ，たまにはいいんでないのこんな質問も。できれば，某新聞で中島らも氏がやっている「明るい悩み相談室」みたいなのを，このページに設けてもいいなぁ，などと考えている今日このごろ。ちなみに，畦地君へのお答えは「極めて正常です」。

◆マクドナルドが1兆円達成記念といって，390円セットを360円に値下げしたようですが，Ｘ68000も何万台突破記念とかいって，Ｘ68000を値下げしたりはしないのでしょうか。

中島　慶　(17)　千葉県

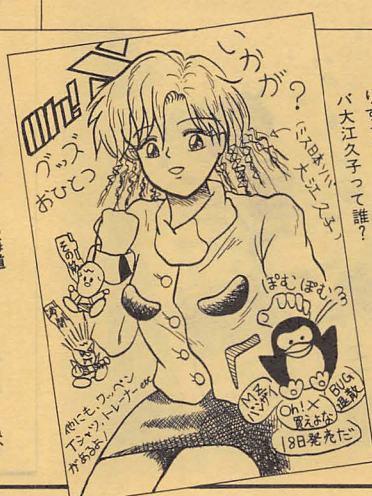
そうですね，10万台突破記念のときには，ぜひともディスプレイとセットでご奉仕価格の10万円……，なんてことがあるわけないか。

◆最近，Ｘ68000のソフトが増えてきましたが，それに従って手抜きのソフトが多くなるのでは，と心配です。うまくいえませんが，かつてのアタリ，ファミコンのように質の悪いソフトによって，Ｘ68000自身の評価が落ちることを考えてしまうと……。Ｘ68000にはいつまでもほかとは一線を画しているといった，高級なイメージを持っていてほしいのです。

安藤　淳二　(19)　京都府

確かに，なんでもかんでもレベルを考えずに発売されるようになってしまうと困るけど，そういったソフトって絶対にＸ68000のソフトのなかでは違う意味で目立ってしまうはず。だからこういったものには，みんなで確かな選ぶ目を持ってさえいれば，そのうちきっと自然淘汰されると思うんですけどね。

◆以前，中森章氏がフロントエンドプロセッサしかほめていなかった「Shogun」を買ってしまった。前評判どおり，変換中には感性が磨け，

◀中島　奨　(22)　北海道
おっと、一瞬手書きのようだけどCGなんですね、これって。ケント紙に熱転写で打ち出したんですかぁ?　こんど私も試してみよっと。

◀山田　純二　(19)　神奈川県
このペンギンくん、Oh!MZに載っていたのよりずっとカワイイですね。ところで、ミス日本ソバ大江久子って誰?

油断してデリートキーを押しっぱなしにしようものなら，キーバッファのダムが決壊して，削除が嵐のように溢れ出る。ちなみに「Master of Monsters」は「SUPER大戦略」より格段に速くなり，磨けるのはせいぜい歯ぐらいのものである。

駒田　常明　(23)　宮城県

◆X1のマニアタイプのころからパソコン歴は5，6年になりますが，新しいパソコン仲間ができてもマシン語の入力方法も知らないような人が多すぎます。どうか，ここでひとつマシン語の入門講座を開いてみてはどうでしょうか。

椎名　劉隆　(17)　千葉県

椎名君のリクエストにお応えして，来月はアセンブラを中心としたマシン語の特集です。お楽しみに。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集室では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲　間

★クラブ「TURBO愛好会」では，X1/X1turboのディスクユーザーを対象とした会員を募集します。主な活動は，ソフトの情報交換や会報の発行などです。BASICやFM音源を知る人，知らぬ人。ゲーム狂の人。どなたでも結構ですからこのクラブに入ってみませんか。会員になりたい方は，ハガキで連絡を。折り返し入会案内書，その他を送ります。　〒032　岩手県久慈市栄町32-114　吉田順　(17)

★X1ユーザーを対象としたクラブ「F.C.G」では初心者を対象とした会員を募集します。活動は，FM音源関係のミュージックや自作RPGシナリオコーナーなど盛りだくさんの内容の会報発行を中心に行っていきたいと思っています。興味のある方は60円切手を同封のうえ封書にて連絡を。　〒751　山口県下関市下関東郵便局私書箱第7号「F.G.C」係　代表者　伊東厚志　(17)

★「Ka-Bi Killers」では，X1turboユーザーで初心者の方（パソコン歴1年未満）で，14～20歳くらいの方を対象とした会員を募集します。現在，活動はソフトの情報交換やオリジナルソフトの発表などを行っています。会費は入会金200円，会費3カ月800円です。入会案内書ご希望の方は60円切手同封のうえ封書にて連絡を。　〒491-01　愛知県一宮市浅井町大日比野南若栗13　市川高久　(17)

★「GAME産業クラブ」では，X1turboでゲーム，システムなどの共同開発を行えるスタッフと，アセンブラ等でゲームを作れるようになりたいと思っている方を会員として募集します。会員の方にはスタッフが作成したユーティリティや実用プログラムが入ったディスク会報を配布しています。興味のある方は60円切手同封のうえ，封書にて連絡を。　〒581　大阪府八尾市南木ノ本2-13-63　庄井美章　(18)

★「X1たーぼCLUB」では，X1turboユーザーの会員を募集します。活動内容は主にゲームの情報交換とディスクによる会報の発行です。ゲームの好きな方，プログラミングの好きな方，男女を問わず参加してください。入会金は500円，会費は300円です。　〒862　熊本県熊本市湖東3-6-5　村井友治　(18)

★「68総研」では，X68000を中心に実用ソフトの情報交換や自作ソフトの交換，ハードウェアの解析・制作などを積極的に行っています。興味のある方はぜひ，ご連絡ください。また各地で活動されているサークルの方々とも親交を深めたいと思っていますので，活動を現在なさっている方もご一報を。　〒590-02　大阪府和泉市光明台1-8-10　吉宮秀幸　(20)

★今度，X68000ユーザーを対象としたユーザーズクラブを創設したいと思うので会員を募集します。活動内容はゲームの情報交換やプログラムの共同開発を予定しています。興味のある方はぜひご参加ください。連絡は往復ハガキで。　〒899-02　鹿児島県出水市中央町516-2　杉川晴彦　(30)

★「倶楽部FIGHTING」では会員を募集します。活動は「なんでも情報誌」の発行が主体です。従ってなんでもやっています。楽しい倶楽部なのでおじさんから女性の方まで，ふるってご参加ください。文章を書きたい方やイラストを描きたい方も大歓迎。入会金は無料です。興味のある方は60円切手と返信用封筒を同封のうえ連絡を。　〒701-42　岡山県邑久郡邑久町みのわ658-1　小谷和典　(18)

売ります

★MZ-2500用RAM-DISK（テレシステムズRM-25E）の未使用品を1万円前後で。連絡は希望価格を明記のうえ往復ハガキで。　〒061-02　北海道石狩郡当別町元町　大竹智樹　(23)

★プリンタMZ-1P17（マニュアル・用紙・リボン付き，箱なし）にMZ-2500用ケーブルを付けて2万5千円で。またX1用FDD・CZ-503F(付属品，箱付き）を1万5千円で。いずれも送料込み。連絡は往復ハガキで。　〒658　兵庫県神戸市東灘区本山南町3-10-30-1108　石川太郎　(16)

★X1用モデムユニット・CZ-8TM1，FM音源ボード・CZ-8BS1，データレコーダ・CZ-8RL1（各付属品付き，完動品)をそれぞれ1万円で。連絡は往復ハガキで。　〒606　京都府京都市左京区高野竹屋町41ハイツ今村105号　佐々木秀紀　(22)

★X1用FDD・CZ-502F（I/F，ディスクBASIC付き）を4万円で。また，漢字ROM・CZ-8BK2を5千円で。いずれも送料込み，箱，マニュアル付き美品。連絡は往復ハガキで。　〒790　愛媛県松山市畑寺1-14-32　成田孝　(15)

★X1用プリンタCZ-80PK（付属品付き，箱なし）を1万円前後で。連絡は希望価格明記のうえ往復ハガキで。　〒529-05　滋賀県伊香郡余呉町坂口626　平野岳志　(17)

★立体映像セットCZ-8BR1の新品同様を，送料込み1万6千円前後で。連絡は往復ハガキで。　〒142　東京都品川区平塚1-8-12　田中達也　(18)

★X1/X68000用RGBシステムチューナAN-8TUの未開封，未使用品を2万円前後で。連絡は希望価格明記のうえ往復ハガキで。　〒630-02　奈良県生駒市小明町1400　吉本健一　(16)

★X1/X68000用24ドットカラー漢字プリンタMZ-1P17（インクリボン，付属品付き）を，送料込み3万円で。連絡は往復ハガキで。　〒305　茨城県つくば市観音台1-7-2　近藤敬一　(17)

★サンヨー電子英和辞書「電字林・PD-1」(和英辞書カード付き)を送料込み1万6千円で。〒252　神奈川県藤沢市長後1243　青木郁夫　(36)

★NECのHiFiビデオデッキVC-N83（付属品，箱付き，完動品）を3万7千円で。連絡は往復ハガキで。　〒051　北海道室蘭市新富町1-6-6　渡辺知巳　(16)

買います

★X1用320KB外部RAMボード，CZ-8EMまたはCZ-8BE2を送料込み各1万円で。またNEW Z-BASICも送料込み1万円で。連絡は往復ハガキで。　〒980　宮城県仙台市北山2-14-4　北山寮　佐藤昇　(20)

★X1用拡張I/Oポート＋漢字ROMをセットで7千円前後で。また，MZシリーズ用バックアップRAM・MZ-1R12を4千円で。いずれも送料別。付属品の有無や状態等を明記のうえ，往復ハガキで連絡を。　〒983　宮城県仙台市田子字堰下17　西谷久範　(20)

★X1用RS-232Cとマウスボード CZ-8BM2（各付属品付き）を，送料込み1万円で。また外部メモリCZ-8BE2を送料込み1万円で。連絡は往復ハガキで。　〒606　京都府京都市左京区北白川上終町6　カーサ鈴村315　石原伸夫　(20)

★X1用ディスプレイCZ-850D/855D/870Dのいずれかを3万5千円で。プリンタCZ-8PC1/2/3のいずれかを2万5千～3万5千円で。また，FM音源ボードCZ-8BS1とカラーイメージボードCZ-8BV1/2を各1万円前後で。各送料込み。連絡は箱，付属品の有無を明記のうえ往復ハガキで。　〒757　山口県厚狭郡山陽町埴生2022-1　柳英一　(18)

バックナンバー

★Oh!MZ1987年6月号(切り抜き不可)を送料込み1,000円で。連絡は往復ハガキで。　〒906　沖縄県平良市西里90　上地悟　(16)

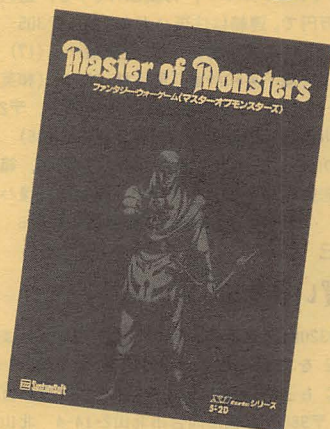
★Oh!MZ1987年6，7月号を送料込み各1,000円で。切り抜き不可，できれば2冊セットで。　〒940　新潟県長岡市上田町1-10　朽木貴広　(16)

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1989年1月18日の到着分までとします。当選者の発表は1989年3月号で行います。

1

システムソフト ☎092(714)6236

Master of Monsters

X1 turbo用5″2D版
3枚組(2ドライブ要)
8,000円 5名

モンスターを率いて世界の覇者になろう！ 魔力と知力で敵と戦うファンタジーウォーゲームを5名の読者に。一大スケールの陣取り合戦が楽しめる。

2

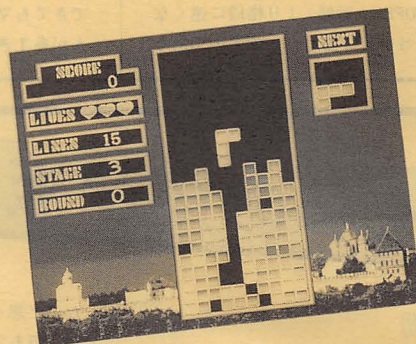
データウエスト ☎06(968)1236

第4のユニット

X68000用5″2HD版
2枚組 6,800円
3名

X68000に第4のユニットが登場。失われた記憶を追っていくうちに次々と起こる事件また事件。ジョイカード対応になったハイパーバトルアドベンチャーをどうぞ。

3

ビー・ピー・エス ☎045(931)0151

TETRIS

X68000用5″2HD版 6,800円 3名

降ってくるブロックを積み重ねてまた積み重ねて積み重ねて……ヒエエ，やめられないよう，というパズルゲーム。シンプルなルールがまた魅力。

4

アスキー ☎03(486)7111

『X68000パワーアッププログラミング』

島田広道，丸川一志，
石橋尚史 共著
B5判，344ページ
2,800円

10名

X68000プログラミング環境をテーマにしたアスキーの最新刊を10名の方に。

11月号プレゼント当選者

1琥珀色の遺言（岩手県）野崎尚行（埼玉県）湯ノ口洋 2ハイドライド3（東京都）八倉巻克巳（大阪府）山根賢一郎（愛知県）伊藤孝博 山川耕司（鳥取県）河上太 3G68K（石川県）天満一裕 4マウスマット（北海道）朝生隆義（新潟県）桜井智（静岡県）竹谷直樹（大阪府）仲谷敏（京都府）小山亮 53.5インチフロッピーディスク（京都府）山下一洋（広島県）藤本秀生 （敬称略）

以上の方々が当選されました。おめでとうございます。品物は順次発送いたしますが，入荷状況などにより遅れることがあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

7カ国語対応電訳機 PA-6110 シャープ

シャープは，Bwareシリーズの新製品として，電訳機PA-6110(14,800円)を12月8日から発売した。

PA-6110は，日・英・仏・独・伊・西・露の7カ国語について会話文や単語を収録しており，ヨーロッパを旅行する人が，外国語で簡単な日常のニーズならこなせるようになっている。

各言語につき会話文約350例および単語約610語が，使用場面に応じ13種のカテゴリーに分けて収められているので，目的の内容を呼び出しやすい。発音はカタカナで表示される。

また，23種類の通貨について円と換算でき，10桁1メモリの計算機能つき。サイズは縦140×横74×厚さ3.5mm，重さは電池を含め約54g。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(260)1161

PA-6110

ハンディデータターミナル RZ-5510 シャープ

バーコードの読み取りとデータ入力が片手で行えるハンディデータターミナルRZ-5510がシャープから発売された。縦180×横75×厚さ20mmの小型タイプで，価格は120,000円から。MZ-6500シリーズを使って，アプリケーションプログラムをCOBOLで開発することができる。

ディスプレイはANKモードで16×4行表示，また電卓機能やメロディでガイダンスするメロディ機能つき。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(260)1161

RZ-5510

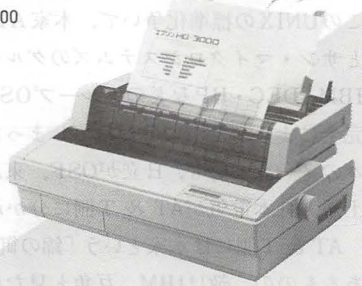
プリンタ新機種 HG-3000/800, VP-135EX, AP-800/550 セイコーエプソン

セイコーエプソンは，インクジェット，インパクト，熱転写の各種プリンタを計5機種，11月から新発売した。

まずインクジェットプリンタとして，136桁のHG-3000(246,000円)と80桁のHG-800(166,000円)。両機種とも約45dBの低騒音設計で，単票・連続紙の切り換えがワンタッチで行えるマルチウェイローディング方式を採用，印字速度はドラフト漢字で220文字/秒が可能である。

コントロールコードはどちらもESC/P24-J84に準拠。

次に，136桁のインパクトドットプリンタVP-135EXは，100,000円という低価格でハイコストパフォーマンスを実現した機種。コントロールコードはESC/P24-J83に準拠しており，印字速度はドラフト漢字の場合80文字/秒が可能で，プルトラクタユニットを標準装備している。

HG-3000

最後に，熱転写プリンタとして48ドット高品位印字を可能にした80桁のAP-800(97,800円)と，同じく80桁24ドットのAP-550(67,800円)。

印字速度は，高速モードのとき，AP-800で漢字53文字/秒，AP-550で漢字80文字/秒。また，微小送り機能によって，印字開始位置を1/180インチ単位で微調整できるほか，通常の黒インクリボンの約5倍にあたる約70万文字分が打てる下書き用マルチタイムリボンの採用により経済性も向上している。

コントロールコードは両機種ともESC/P24-J84Cに準拠。

〈問い合わせ先〉

セイコーエプソン㈱　☎0266(52)3131

インテリジェントモデム PV-A1200MKⅢ アイワ

PV-A1200MKⅢ

アイワは，1200/300bps対応全2重インテリジェントモデムPV-A1200MKⅢを12月中旬から発売する。価格は21,000円。

PV-A1200MKⅢは，自動発着信(AA)型で，CCITT/BELL規格に準拠し，ヘイズATコマンドに対応している。

サイズは幅160×奥行き222×高さ47.5mm，重さは1.2kg。

〈問い合わせ先〉

アイワ㈱　☎03(827)3670

パソコン専用の無停電電源装置 STAND-BY 高岳製作所

停電や瞬間的な電圧低下の際にパソコンなどのデータを守る無停電電源装置STAN

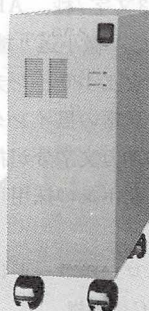
D-BYが高岳製作所より発売された。価格は158,000円。

このタイプは，通常は電力を素通しし，停電時のみバッテリーからパワーを供給するもので，常時は電源の損失はほとんどない。また低騒音であることも特徴となっている。

停電時のバックアップ時間は1.0KVAで8分間。サイズは幅180×奥行き460×高さ480mm，重さ45kg。

〈問い合わせ先〉

㈱高岳製作所　☎03(211)1671

STAND-BY

カードサイズの電子辞書
受験SAY
セイコー電子工業

受験SAY

セイコー電子工業は，カードサイズの電子辞書コノサイズディクショナリーシリーズの新ラインナップとして，電子英単語帳・受験SAYを12月8日から発売した。価格は7,500円。

この電子カードは，大学入試での出題頻度の高い単語5,156語が，出題頻度順に4ランクに分けて収録されたもの。これらの単語には，綴り・発音・アクセント・変化形で注意を要する場合マークがつけられ，それを表示するようになっている。

そのほか重要熟語981語も収録，同・反意語や派生語などの関連語もキー操作で表示できる。以上のデータは旺文社の監修。

縦55×横91mmのコンパクトサイズで，定期入れにも入る。

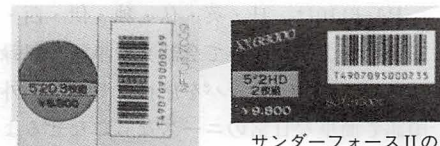
〈問い合わせ先〉

セイコー電子工業㈱　☎03(682)1111

INFORMATION

オリジナルCDプレゼント
テクノソフト

テクノソフトの新九玉伝またはサンダーフォースIIを購入した人全員に，オリジナルCDがプレゼントされる。

応募には，上記のゲームソフトの表紙の一部(写真参照)を切り取って，裏面に住所・

新九玉伝の切り取り部分

サンダーフォースIIの切り取り部分

Again Watch

やや静かだった'88年

1988年は竹下新総理，原発，リクルート，ソウル五輪，ご病気，消費税，米国大統領選挙などなど，実にいろいろなことがあったが，ふりかえってコンピュータ業界，特にパソコン業界では，これといった話題の出なかった1年であった。これは良いことか，悪いことか。

ともあれ，年末年始号の恒例として締めくくりはしておきたい。少々インパクト不足かもしれないが，1988年の10大ニュースをまとめてみよう。

1）IBMと富士通の互換機著作権紛争に決着(11月)

パソコンとは関係ないけれど，今年のコンピュータ業界の最大のニュースはやっぱりこれ。1982年から延々と争われてきたソフト著作権紛争が，米国国際商事仲裁協会(AAA)に委ねられたもの。

結論として，富士通はOSのインタフェイス情報をAAA経由でIBMから今後10年間にわたり入手できるシステムが確立した。ただし，その代償として，富士通はIBMに使用料を支払う。金額は，過去の分が481億円(うち193億は支払い済み)で，今後の使用料は毎年算出するが，来年分は30億円から32億円。高いことは高いが，「公認料金としてはなんとかなる」という感じだ。

2）メモリの品不足が激化(1年中)

1988年をピークにして1Mビットと256Kビットのダイナミック RAM，256Kビットと64Kビットのスタティック RAM，さらにはマスクROMと，メモリが軒並み不足となった。おかげでパソコン，ワープロ，ワークステーションから家電製品まで，さまざまな完成品の生産に支障をきたした。東京・秋葉原でのスポット価格は，定価350円の256K DRAMが1,500円までハネ上がった。半導体メーカーはたいへんな利益を上げたわけだ。

3）UNIX，2派対立へ(春以降)

高級型パソコンといえるワークステーションの主力OSとして注目されているUNIX。このUNIXの標準化争いで，本家AT＆Tとサン・マイクロシステムズのグループと，IBM・DEC・HPなどのグループOSFとに，完全にまっぷたつに分かれてしまった。日本のメーカー勢も，日立がOSF，東芝・富士通・日本電気がAT＆T側に分かれた。

AT＆T側には本家という「錦の御旗」はあるものの，敵はIBM。互角と見たほうが間違いないだろう。いまのところ，両派和睦のメドはまったく立っていない。

4）ウイルス騒ぎ起こる(9月)

日本電気のパソコンネットワークPC-VANで，コンピュータウイルス騒ぎが起こった。実際には，伝染性のないダイレクトメール型の悪質なハッカーソフト「トロイの木馬」ではあったのだが，なんの心配もせずにパソコン通信を楽しんでいた日本のパソコンマニアたちを恐れさせたことは，やはりベストテン入りさせるべきバリューがあるだろう。

5）日本電気，ヤミ再販で公取委から警告(7月)

向かうところ敵なしの，独走を続ける日本電気のPC-9801だが，1986年に発売したPC-9801UVで同社が販売店に対し値引き率の下限を設定，要請したことがわかり，独占禁止法に抵触するかもしれないことを公正取引委員会から警告された。エレクトロニクス製品では極めて珍しい警告で，98がついに天下の公取委から売れすぎのお墨つきをもらってしまった，ともいえる事件だった。

6）AXパソコン発売，三洋電機が一番乗り(2月)

氏名・電話番号を記入し，240円切手3枚を同封して下記宛に送付する。

〒857 長崎県佐世保市福石町4-14

(株)テクノソフト　CDプレゼント係

締切りは1989年1月31日，発送は3月中旬。また上記ソフトに使用されている曲をアレンジするか，またはゲームのイメージに合う曲を作成して送ると，優秀作品は同CDに収録される。応募先はプレゼントと同じ。ディスクまたはDATに入れて送付すること。

〈問い合わせ先〉

(株)テクノソフト　☎0956(33)5555

ゲームミュージックの新企画
ファルコム・レーベル誕生

ザナドゥ，イース，ソーサリアンなど，最近のゲームミュージックにはゲーム本体より話題になるものが少なくないが，このほどソフトメーカー日本ファルコムによるレコードレーベル，ファルコム・レーベルが誕生。その発足記念として，「ファルコム・スペシャルBOX'89」が12月5日にキングレコードから発売された。

ファルコム・スペシャルBOX'89

8cmCDのMINIアルバム6枚入りで，定価7,800円。ザナドゥ，イース，ソーサリアンなどの曲を交響曲やボーカル，プラズミックス，ディスコサウンドなどにアレンジしたものを収録しているほか，MJQのD.マシューズがプロデュースした1枚も話題となっている。

〈問い合わせ先〉

日本ファルコム(株)　☎0425(27)6501

BOOK

X68000パワーアッププログラミング
アスキー

X68000でプログラミングを楽しんでみたいというユーザー向けに，『X68000パワーアッププログラミング』が発売された。ハードやソフトの基本からOS，BASIC，X BAS to CまでX68000のプログラミング環境について詳しく解説している。

島田広道，丸川一志，石橋尚史 共著

B5判，344ページ，2,800円

〈問い合わせ先〉

(株)アスキー

☎03(486)7111

X68000パワーアップ
プログラミング

ニュースベストテン'88年　1989-1

1987年秋に規格が決まったAXパソコン。下馬評どおり，三洋電機が一番手として2月末にMBC-17Jのサンプル出荷を開始した。続いて三井物産(アルプス電気のOEM)，三菱電機，シャープが加わり夏から本格的に販売が始まり，沖電気，カシオ計算機，京セラ，キヤノンなども参戦，10社以上のメンバーになった。

ご覧になってお気づきのように，Bリーグ以下の顔ぶればかりである。しかし，MS-WINDOWSを標準装備し，最新鋭ハードで武装したAXパソコンはなかなか強力なマシンである。IBMの英語版ソフトもバカにならない。強力な勢力となる可能性は高い。現実に三菱電機のラップトップMAXYは人気があるのだ。

7）TRONチップ，いよいよ登場(7月)

待っても待っても出てこないのがBTRONパソコンだが，その間隙をぬってTRON方式のマイクロプロセッサTRONチップが完成してしまった。最初に登場したのは7月。日立が開発したGマイクロ/200である。公表性能6MIPSの32ビットプロセッサで，額面どおりだとi80386よりも断然速い。来年春には富士通が上位製品のGマイクロ/300を，三菱電機が下位のGマイクロ/100を完成させる。宗教視されているTRONだが，ハードの心臓部が先にできてしまったことはプロジェクトの順調さを物語っているのではなかろうか？

8）98，150万台を突破(4月)

1987年3月に100万台を突破したPC-9801だが，わずか13カ月で早くも50万台を上乗せした。売れ行きペースは速まるばかり。98の独走はとどまるところを知らない模様。1988年もUV11，CV11，LV21，RA2/5，RX2/4，VM11，LS2/4と新製品が相次いだ。

9）Next登場(10月)

Apple Computer社の創設者であるスティーブン・ジョブズ氏が，同社を退職後ようやく行動を開始した。新会社Next社を設立，第1号商品としてワークステーションNextを発表したことはご存じのとおりである。Nextは，モノクロメガビットディスプレイに光ディスクドライブで武装した32ビット高性能製品。今月号の本誌でも，有田隆也氏が知能機械概論で使ってみたいとラブコールしている。

「さすがにすごい」，「ありきたりだ」と賛否両論ながら，ビジネスウィーク，ニューズウィーク両誌の表紙に登場する話題性はさすが。

10）トムキャット互換機が登場(2月)

マルチ互換機を実現するトムキャットコンピュータ社製のパソコン互換システムVSLを，ソフトハウス団体の日本パソコンソフトウェア協会が全面的にバックアップすることを決定。ハードメーカーとトムキャットとのブッキングを同協会系共同出資会社が行った。テストとしてトムキャット自身が，VSLを使った98/AT互換機を開発，販売した。話題性は十分だったが，採用メーカーは精工舎が11月に名乗りを上げたにとどまった。

トレンド総評

さて，1987年は互換機と98対抗機を軸にした標準化関係の話題が多かったが，1988年は一転して社会部ネタと海外ネタが大部分を占めてしまった。とくに，パソコンがパソコンだから話題になるのではなく，汎用機，ワークステーションとの境のワクがなくなって「コンピュータ」として扱われてきたことは見逃せない。いよいよパソコン業界も，本当の意味でコンピュータ界のひとつとして正式に認可され，国際化，社会現象化してきたのだろう。1990年代への前哨となる次の1年はいかに？　(K.T.)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーは，本誌年間モニタの方々のご意見を紹介しております。今月は11月号の記事に関するレポートです。

●「プリンタ用外字登録ツール」は，自作年賀ハガキプログラムなんか使うときに便利でしょうね。本来こういうツールは標準でついていてもおかしくないと思います。私は，1986年のOh！MZ 4月号で発表された拡張漢字BASICを使っていました。現在はX68000ユーザーですが，XIGを使っている妹に勧めてもいいですね。それから，「Z80マシン語ゲーム工房」はとてもやさしく思えます。なぜでしょうか？ それは1つひとつのことが単純だからです。その集まりを一度に見ようとするから難しくなってしまうんですね。わかってくると読んでいてとても楽しくなります。

八木　信彦（20）XIG, X68000, PC-1245　愛知県

●「グラフィックのモノクロ出力方法論」は面白かったです。特に濃淡表現の2値化する部分が。これならランダムディザなどと違って結果に対して再現性がありますね。プリンタに関しては，ESC/Pが提唱されたとき，これでプリンタコードが統一されるかなと思ったけど甘かった。なかなかうまくいかないものですね。今度そのあたりのことにも触れてください。また，「拡大文字のスムージング」に書いてあったベクトル情報の抽出についてですが，24ドット文字からベクトルデータを完全な形で取り出すプログラムなんてできたら，それだけで卒論として通用するような気がします。

中島　奨（22）MZ-1500, PC-9801VX, PC-1360K　北海道

●外字登録は結構ややこしくてなかなかやる気が起こらないのですが，「プリンタ用外字登録ツール」のようなものがあるとたいへん便利です。また「グラフィックのモノクロ出力方法論」は，今まで考えたこともなかったモノクロのハードコピーの方法で，ユニークで利用価値も高いと思います。

安本　威一朗（14）MZ-2000, X1turboII, PC-6001　兵庫県

●「制御コードは攻めの基本」はなかなか役に立った。これを参考に方眼紙を印刷するプログラムを作ってみたところ，テストで狭い範囲に印字するときは気にならなかったスピードが，B5判用紙いっぱいに印字したとき，あまりにも遅くなるので驚いてしまった。所要時間なんと40分。そんなに待っている自分もどうかと思うが，やはりビットイメージはマシン語かなとも思ってしまった。でも遅いとはいえ方眼の大きさだって指定できるし，けっこういけるじゃないかとも思う。

藤崎　和義（17）X1turboZ　東京都

●これまであまりドットプリンタに縁のなかった僕には，プリンタの実情と基礎を理解するうえで「メカニズムを理解しよう」は役に立ちました。また，せっかくカラープリンタがあっても文字を打ち出すだけではもったいないので，「文字と図形の混在印字」のビットイメージなどは避けて通れないでしょう。「オリジナル印刷キットを作ろう」はいくつものサンプルプログラムを紹介してあってなかなかいいと思います。作者の言うとおり，こういったものは紙や字の形，配置などに思いきりこだわらないといいものは作れないと思います。これをもって，「パソコンは役に立つ」と実感してみるのもいいのでは。また，「Z80マシン語ゲーム工房」についてですが，僕は説明の量が適当だと思います。あまり説明が細かいことなく自分で考えてみることを促されるので，プログラムのテクニックなどがよくわかります。ただ，いまのところ分量が多くて追いつくのに必死という感じですが。

星　大地（15）MZ-700, PC-1475　静岡県

●今回の「Z80マシン語ゲーム工房」では，ゲーム，特にアクションに欠かせない仮想V RAMの使い方でしたね。ELFESⅣでも使ってますが，ここでの処理，特にグラフィックのときは重ね合わせがゲームの見せ場でもあります。全体的によくわかりましたが，クリッピング処理についても書かれてあったら良かったのではと思いました。

田端　勝也（19）MZ-2500　石川県

●「グラフィックのモノクロ出力方法論」の1ピングラフィックを見て，新聞写真や点描を思い出しました。でも，点を打つ打たないでこれだけ描けるなんてちょっと驚いてます。ただ，印字方向が横なので横方向の相関はうまくいっても，縦の相関が落ちてしまうのでは，という気がしましたが，考え方としては相関の方向に一定の方向の相関を加えて印字しているといえるので，全体の相関はただ平行移動しただけということなのでしょうか。それから，「オリジナル印刷キットを作ろう」でちょっと意外だったのがディスクエンベロープの作成。いつもできあいのものを使っていて，自分で作るという発想がなかった。プログラムを読んで，このワンポイントエディタを自分の機種でも使えるようにしてみようかと思っています。

金田　敦（25）PC-9801VM2　東京都

●マルチタスク環境というのを僕が初めて見たのは，AMIGAが発表されたときです。あのきれいなグラフィックとスプライトでマルチタスクしている画面写真を見て「いいなあ」と思ったものです。そしてX68000用のOS-9が出るわけですが，僕らにも手の届く値段であればありがたいですね。OS-9を買えたとしても「タイマしか動いていないシステム」なんてシャレにもなりませんから，特にCなんか高そうだな。

橋本　浩二（17）X1F model 10, X68000ACE-HD　兵庫県

ごめんなさいのコーナー

11月号　STAR TREK

P.118の2160行の“v4l”は“val”の誤りです。また，P.133の16820行と16850行の，

if a$＝“☆” then～

を

if a$＝“☆” or a$＝“R” then ～

のように訂正してください。

12月号　ピコマゲドンへの道

P.51　ゲーム開始後，バスエラーが発生するという症状がありました。63，73行の，

for (i＝0; i<mlast～

という部分を，

for (p＝i＝0; i<mlast～

に変更してください。

12月号　SOURCERY

P.64 “LD B,H : LD C,L” が “LD BC, HL” になるなど，一部のマクロ関係の部分が誤動作していました。リスト1の修正を加えてください。

12月号　特集 美しい響きの要素とは

P.95 参考文献2,3の著者名に不備がありました。正しくは「石黒脩三」です。お詫びのうえ訂正いたします。

リスト1

```
38DD FE 40 DA DF 33 FE 70 D2 : 6A
38E5 DF 33 E6 07 FE 06 D2 DF : B4
38ED 33 4E 23 CD 58 40 CA DF : B2
38F5 33 7E A9 FE 09 C2 DF 33 : 35
38FD A1 28 05 FE 09 C2 DF 33 : A9
3905 CD 69 40 4C 44 20 20 20 : 66
390D 00 79 0F 0F 0F 0F E6 03 : 9E
3915 47 79 0F E6 03 4F C3 5D : 27
391D 3B                      : 3B
----------------------------------
SUM: 33 C2 EF F0 F1 46 93 76 3A06
```

バグに関するお問い合わせは
☎03(263)2230(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

求む！スタッフ 1989を支えるのは君たちだ

▼1989年，いよいよ'80年代最後の年。Oh！Xの年頭教書は，'90年代に向けて一層の誌面充実を図るため，スタッフ募集で始まります。

仕事の内容は主に原稿の執筆，投稿のチェックなど。基本的に不定期な作業ですから，決められた時間の拘束などはありません。応募資格は，東京近郊にお住まいの大学/専門学校生，または社会人でパーソナルコンピュータに興味を持ち，Oh！Xのスタッフとしてがんばろうという気のある人。特に使用機種は問いませんが，MZ，Xシリーズのユーザーであればなお結構です。

希望者は，氏名・住所・電話番号（あれば必ず明記）・略歴に加え，得意分野や自己PRなどを含めた自由論文をレポート用紙2枚程度にまとめてOh！X編集室スタッフ募集係まで送付のこと。なお，投稿経験があればそれも記してください。

▼というわけで，1988年度GAME OF THE YEARノミネート作品の発表です。絢爛たるこのページを作るために，担当とスタッフの面々は朝から晩まで過去1年分の愛読者カードに埋もれ，ノミネート作品の選出にがんばってくれました。その努力を無駄にしてはいけない！　推薦するソフトとそれに対するコメントを書いて，どんどん編集室まで送ってください。皆さんの意見を批評を提案を，できるだけたくさん載せたいと思います。それがパソコンゲームを育てることにもなるわけですから。応募は特に愛読者カードには限定しません。詳しくは26ページをどうぞ。4月号の発表が待ち遠しいなあ。おっと，その前に編集室は再びハガキの山に埋もれる運命にあるのだった，うーん。

▼「日曜日が待ち遠しい！」というのはフランス映画のタイトルですが，「日曜日がうっとうしい！」というのは年末スケジュールの立て込む編集室の悲鳴です。つまり，へたに休日があるとその分仕事が遅れてしまうという悲しい意味です。でも，日本全国の労働者は，皆同じ条件で働いているわけですからね。クリスマスにお正月と続くイベントを楽しみに，皆さんがんばりましょう。

受験生の方は追い込みですね。健闘を祈っています。

●投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh！X「㋢㋮㊔」係

SHIFT・BREAK

▶先日，アメリカでステルス爆撃機の機体が報道陣の前で公開された。テレビでその姿を初めて見たときの印象は「なんか，ひと昔まえにはやったゲイラカイト（西洋凧の一種）みたいだな」。レーダーに探知されないように設計されたとはいえ，まさかあんな形にするとは。妙なことで改めてアメリカ人の独創性を再認識してしまった。　(R.K.)

▶次はコンチネンタルサーカスにメットを被って乗ろうかな。本命はチェイスH.Qのポルシェ928 S4なんだけど。ちはみにX68000のスペハリも持ってます。　借地人(S.K)
（こらこら，私のスペースを4行も貸したんだからちゃんと笑いを取らなくちゃだめだよ，金子。君がお笑い3人組のリーダーでしょ。　大家さん(で)）

▶この号が出る頃にはメリークリスマス＆ハッピーニューイヤーなのであるが，冬といえばスキーだっ！今年は気合いを入れて，4Sケブラー＆957＆ライケルとおごったので初滑りが楽しみ。しかし問題は山積み。クリパー，締切，金欠。はたして年を越せるのか？　でもNさんごめんなさい。私は12月8，9日枡立高原にいます。　(C.W.)

▶セカンドマシンとして「高速な8086」が載っかったパソコンを購入したわけだ。生意気にもメインマシンであるX68000にも付いてないハードディスクを繋いでワープロ＝原稿書きに使う予定である（と言いながらturbo Cを買ってしまう僕の性よ）。X68000にもっとましなワープロがあればこんな出費はしなくて済んだところなんだが，ね。　(Mu)

▶日本のあるところに伝わる古史古伝のひとつに，秀真伝というのがある。これはホツマツタエと読む。イースター島の伝説にホツマツアというのがある。日本とイースター島。何か関係があるのだろうか。出雲大社と伊勢神宮を結ぶ直線をずーっと伸ばしていくと，イースター島に辿りつくという。古代の出来事は誰も知らない。　(K)

▶最近朝早くまでゲームをしているので，お気に入りのキングスシンガーズのカセットでは起きられなくなってしまった。そこでX68000のテレビで起きようと思い，今日編集部でやり方を教わった。うーん，いっそのことBASICを立ち上げて音量やチャンネルをガチャガチャいじくってやろうか。朝からパニックで起きるとさぞ疲れるだろうねえ。　(K.S.)

▶X68000版のTETRISには期待していたのだが，ステージが変わっても背景とBGMが同じなのでがっかりした。最高点をディスクに保存できないのも，大勢で最高点を競い合うという点から見れば辛いものがある。ゲームのでき自体は結構いいのだが，Mac版を知ってしまった私にはいまいち熱くなれないのが悲しい。　(KO)

▶ああ，年の瀬だぁ。この時期になると頭が痛いのが帰省のための航空券なのです。外国と比べて馬鹿高い料金を取っておきながら，需要に応えられないという大ボケ航空行政は一体どうなっているのでしょうか。パンナムでもノースウエストでもいいから，圧力をかけて日本市場に参入してくれないもんですかねぇ。こういう進出なら大歓迎なんですが。(M)

▶細菌学を専攻していた友人の下宿で冷蔵庫を開けると，「ボツリーヌス」と書かれたペトリ皿がいつも鎮座ましましていた。貴重な食料を勝手に荒らしていく友人(私じゃない)対策だといっていたが，かつての同級生と結婚した今でも続けているらしい。夜中に食い散らかすご亭主対策とか。まったく10年前から成長しないカップルだ。　(よ)

▶HEシステムにメガドライブとマシン室はゲーセンの装いの今日この頃。昔，うちの兄貴の友だちが逆輸入でしか手に入らないというでっかいバイクに乗っていました。それは国内ではほとんど見られないという6気筒のバイクでした。そーかー。CBXとCBか……。というわけで，CB-1100Rは4気筒でいいんだそうです。ごめんなさい。　(U)

▶12月号の愛読者プレゼントのなかで，写真撮影に使われた福袋をぜひともくださいという方が多いのには驚きました。実はあの袋は，うちの（よ）嬢が参考書（？）を見ながら，せっせと作ってくれたものなのです。だからそう簡単に人はあげられません。だって，彼女の唯一の嫁入り道具になるかもしれない品物なんですからネ。　(N)

▶昨年の暮れは「インクリボンが品切れで年賀状が出せなかった」という話が多かったけど，今年は大丈夫だろうか？　いくらZ's STAFFやPrint Shopがあってもリボンがないとお手上げですもんね。さて，1月の15日ぐらいまでに編集室に届いたカラーのイラストやCGは，3月号でまとめて紹介の予定ですから皆さんよろしくお願いします。　(T)

microOdyssey

硬く弾む靴音が長い回廊にこだまする。空気は冷たく，しかし運ばれる緊張と興奮は，続く波乱を容易に予想させるかのようだ。手にした知らせを一刻も早く伝えようと，靴音の主は足を速め，やがて目的のドアに飛び込むと居並ぶ人々に向かい声高に叫ぶ。

“It's name is Sputnik！”

こうして，米ソ両超大国による宇宙開発競争は始まった。

地球を外から見ることで機先を制したのはソ連だったが，やがてケネディ大統領の言葉どおりアポロ計画が人類を月へ運び，1970年代からはそれまでの使い捨てロケットに代わりスペースシャトルが登場した。そして1986年1月，チャレンジャー号が爆発する。

以来3年近いブランクを経て，1988年9月，NASAはスペースシャトル計画の復活をかけたディスカバリー号の打ち上げに成功した。

数々の実験を終え，TDRSの打ち上げも済ませて無事帰還したディスカバリーだったが，国内外でのスペースシャトル計画に対する批判はそれで払拭されたわけではない。

9月の打ち上げ成功で，NASAは人間を宇宙に送る能力が健在であることを証明してみせた。しかしシャトル計画当初のように，スペースシャトルが「安価で信頼のおける宇宙の輸送手段」であるとは誰も考えていない。国家宇宙輸送システムをシャトルに一元化する方針も，シャトルによる商業衛星の打ち上げ計画も，みな著しく後退した。なかでも目立つのは有人飛行に関する批判である。シャトルから通信衛星を打ち上げる，そんな20年前のロケットでも可能だった目的のためになぜ人命を危険にさらす必要があるのか。NASAは宇宙で一体何をするつもりなのか。

確かに，こうした批判は正しいのだろう。シャトル計画は高価な過ちであり，第一級の政策ミスだといわれるのも，人的資源やコストの膨大さを考えれば当然出てくる意見だと思える。NASAが，人々の熱狂と支持を得るために有人飛行が必要だと思い込んでいたというのも確かだろう。しかし，事実人々は熱狂してきた。

複雑な事情に取り組む関係者が聞いたら怒るかもしれないが，人間が宇宙に出ることそのものに，私たちはまだ熱狂していていいと思う。新しい世界を開くときのエネルギーの源はそこにあると思うからだ。

映画「2010年」で印象的だったもののひとつに，名優ジョン・リスゴー演じるカーノウ博士が初めて宇宙空間に足を踏み出す場面がある。彼は，ロシア語で「チキン」を意味する言葉を一生懸命つぶやきながら，HALを積んだディスカバリー号に乗り移る。モノリスの秘密に迫ったときより，また彼らが地球に戻る算段をしているときより，手に汗握ったシーンだった。

自分の足の下に何もないという事実。それがどれほどの恐怖をもたらすのか，残念ながら私には想像もつかない。しかしその恐怖を味わうためなら，人間は皆，どんな犠牲でも厭わないだろうと思う。

そうして，かつてスプートニクの名に興奮した靴音の主は，人類初の有人火星探査船の名を知ったとき，やはり同じように走るのだろうか？

いずれにせよ，それは少し先の話である。(よ)

バックナンバー常備店

地域	地区	店名・電話
東京	神保町	三省堂神田本店5F 03(233)3312
	〃	書泉ブックマートB1 03(294)0011
	〃	書泉グランデ5F 03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660
	八重洲	八重洲ブックセンター3F 03(281)1811
	新宿	紀伊国屋書店本店 03(354)0131
	高田馬場	未来堂書店 03(200)9185
	渋谷	大盛堂書店 03(463)0511
	池袋	西武百貨店11Fブックセンター 03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(981)0111
	町田	久美堂東急ハンズ店 0427(28)2783
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	西武百貨店10Fブックセンター 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は，最寄りの郵便局にある払込用紙に，

口座番号　東京1-29307

加入者名　株式会社日本ソフトバンク

とご記入のうえ，**年間購読料6,500円**を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!X定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

Oh!X 1月号
■1989年1月1日発行　定価540円　■発行人　孫　正義　■編集人　笹口幸男
■発行元　(株)日本ソフトバンク
■出版事業部　〒102 東京都千代田区九段南2-3-26 井関ビル　☎03(261)4095　FAX 03(262)8397
編集室☎03(239)4156
出版営業☎03(261)4095
広告営業☎03(297)0181
■本　社　〒102 東京都千代田区九段南2-3-14　靖国九段南ビル　☎03(263)3690㈹
TELEX 東京　232-4614JSBTYJ　FAX 03(263)3660
■西日本営業部　〒541 大阪府大阪市東区南本町2-6　明治生命堺筋本町ビル10F
☎06(264)1471㈹　FAX 06(264)1481
■印　刷　凸版印刷株式会社

超高速の橋出現！

X1・X1turbo用
SUPER DEVICE MONITOR "T"

BLUE SKYはコンピュータ通信にオブジェクトデータの橋を架けました。今迄はRS-232Cでオブジェクトデータを通信する時は，アスキーデータに変換して行っていたコンピュータ通信を，直接オブジェクトデータのままで，しかも，特殊なデータ圧縮を施して，今迄にない超高速で通信する事が出来るX1turbo用の『SUPER DEVICE MONITOR "T"』を開発しました。既に好評発売中のMZ用の『SUPER DEVICE MONITOR "T"』とはRS-232Cにより双方向の超高速通信が出来ます。

エディット機能も呼び出したセクターを豊富なコマンドを使ってワープロ感覚で自在に変更・書き込み等のデータの編集が簡単に出来ます。アクセス出来るディバイスもハード・ディスク，MS-DOSやX68000で使用しているフォーマットの2HDのディスクなど各コンピュータに接続された殆どのディバイスをエディットする事が出来ます。

★任意のディバイスから他のディバイスへセクター単位で高速転送が出来る。

★任意のセクターをほぼ瞬間的に縦・横チェックサムとキャラクターダンプ付き表示が出来る。

★エディット機能はワープロ感覚で表示したセクターのオブジェクト・データを1バイト単位で変更・複写等多彩なエディット機能を備えている。

★turbo内のBIOS用ROMやturboZII標準装備の内部増設メモリーにも直接アクセス出来る。（turboのみ）

★任意のディバイスの複数のセクターを他のディバイスと比較・照合が出来る。

★キャラクターダンプは漢字の表示も出来る。（X1は除く）

★RS-232Cのボーレートの変換はボタン一つで切り替えられる。

★X1フォーマットやMZフォーマットのディスクがアクセス出来る。

★X68000やMS-DOSフォーマットのディスクにもアクセス出来る。（turboのみ）

★255バイト迄のデータを任意のディバイスの複数のセクターから検索する事が出来る。

★キャラクターダンプで表示出来る漢字には区点・JISの表示も出来る。（turboのみ）

★2HD及び2DDのディスクもアクセス出来る。（turboのみ）

★RS-232Cを使っして他のコンピュータとの間で相互に特殊なデータ圧縮法に因り複数のセクターのオブジェクト・データを通常の最高32倍（理論値）の超高速での転送が出来る。（X1は除く）

SUPER DEVICE MONITOR "T"	X1	5″	2D	10,000円
（turbo用の2HDは受注生産）	X1turbo	5″	2D／2HD	13,000円
	MZ-2500・2800	3.5″	2DD	13,000円

ロードに長時間かかる多分割のテープ版のゲームがボタン操作一つで何本も1枚のディスクに整理が出来て表示したリストから遊びたいゲームを指定すると一瞬でロード出来る『EXTRA HYPER＋α』もあります。

EXTRA HYPER ＋ α	X1・X1turbo	3″・5″	
	MZ-2000・2200・2500	3.5″・5″	各14,000円

BLUE SKY Co.

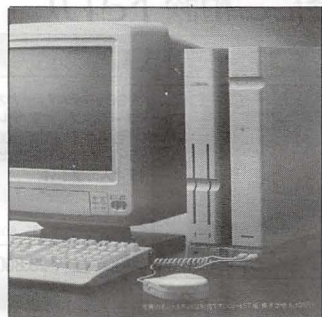
▶お求めは全国の有名マイコンショップでどうぞ。
通信販売をご希望の方は当社へ直接、商品名・機種名・メディア名・住所氏名・電話番号を明記の上、現金書留にてお申し込みください。（送料無料）

株式会社 BLUE SKY
〒411 静岡県三島市加茂16-4
☎0559-72-6710

OPEN8周年記念特別企画

キャンペーン期間
12/1〜1/31まで

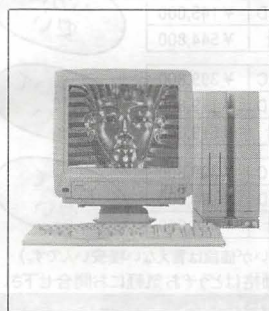
ビックリ特別企画!!

当社ソフト開発用におろしたX68000(CZ-600C)をACE HDに変更するためX68000ファンに特別に超特価でおわけ致します。限定8台

CZ-600C
本体のみ
¥189,000
送料¥1,000

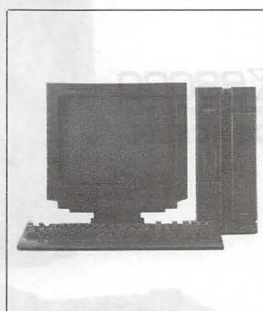
10MBハードディスク(2台)	¥58,000
20MBハードディスク(3台)	¥78,000
VP-135PC (2台)	¥68,000

その他1MBメモリ、コプロボード、GPIBボードあり

長期クレジット　頭金9,000 ¥6,400×36回

BASIC HOUSEおすすめ特別セット

Hyper COBURA SET A

CZ-601C	¥319,800
CZ-603D	¥ 84,800
ハンデープリンタ	¥ 24,800
メロディBOX	¥ 16,800
当社オリジナルソフト(7本)	¥ 74,400
5インチ2HD10枚PDSソフトサービス	
定価合計	¥520,600
超特価	¥389,000 〒1,000

長期クレジット　頭金9,000　12,800×36回

Hyper COBURA SET B

CZ-611C	¥399,800
CZ-603D	¥ 84,800
ハンディープリンタ	¥ 24,800
メロディーBOX	¥ 16,800
当社オリジナルソフト(7本)	¥ 74,400
5インチ2HD10枚PDSソフトサービス	
定価合計	¥600,600
超特価	¥459,000 〒1,000

長期クレジット　頭金9,000　15,200×36回

BASIC HOUSEオリジナル

X68000シリーズ

B6-6301	BASIC拡張関数パッケージ	¥ 9,800
B6-6302	CP/Mエミュレータ	¥ 19,800
B6-6303	アイコンエディタ	¥ 4,800
B6-6304	ディスクキャッシャー	¥ 6,800
B6-6305	C言語ライブラリ	¥ 6,800
B6-6306	BASIC拡張関数パッケージC言語ライブラリ付	¥ 14,800
B6-6307	Toys & Tools	¥ 6,800
HANDY PRINT jack		¥ 24,800
MELODY BOX	MIDIインターフェース	¥ 16,800
KGB-X68ADC	16ch12ビットA/D変換ボード	¥128,000
KGB-X68PIO	アイソレーション16Bit入出力ボード	¥ 68,000
KGB-X68UNB	ユニバーサルボード	¥ 6,800

MZシリーズ

B7-2501	PC-8801→MZ-2500	テキストコンバータ	¥ 3,000
B7-2502	PC-8001→	〃	〃
B7-2503	PC-6001→	〃	〃
B7-2504	FM-77 →	〃	〃
B7-2505	MSX →	〃	〃
B7-2506	S1/L3 →MZ-2500	テキストコンバータ	¥ 3,000
KGB-MZ1	超低価格計測制御ボード		¥ 15,500

X1/X1turboシリーズ

KGB-X1S	低価格アナログデジタル入出力ボード	¥ 19,800
KGB-HDIF	X1turbo専用ハードディスクインターフェースボード	¥ 16,000
KGB-PIO	高級絶縁型パラレル入出力ボード	¥ 42,000
KGB-AD12	高級16ch12Bit A/D変換ボード	¥118,000
KGB-DA4	高級4ch12Bit D/A変換ボード	¥ 98,000
KGB-488	GP-IBインターフェース(マニュアルソフト付)	¥ 58,000
B6-3301	PC98↔X1turbo相互ファイルコンバータ	¥ 4,800

PC-8801/PC-9801シリーズ

KGB-PC1	KGB-MZ1のPC-8801版	¥ 15,500
KGB-98S	PC-9801シリーズアナログ入出力	¥ 19,800
	デジタル入出力ボード(D/A付)	¥ 25,000

Macintoshシリーズ

PRINT jack	プリンタドライバー	¥ 38,000
MOUSE jack	マウスドライバー	¥ 4,800
MELODY BOX	MIDIインターフェース	¥ 19,800

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研　本社営業部／マイコンショップ／通販部　宇都宮市竹林町503―1　TEL0286-22-9811　FAX0286-25-3970
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

秋葉原価格でローンができます
電気の街秋葉原で 24年の信用!!

AVCフタバ
AUDIO VIDEO COMPUTER

☎03(253)7661

至新宿 電波ビル リビナ 至神田 中央通り 秋葉原駅 ヤマギワ 至上野 日通ビル 外堀通り
AVCフタバ電機通信販売部
AVCフタバ電機本社

AVCフタバ電機
〒101 東京都千代田区外神田2-9-8
神田ユニオンビル ☎03-253-7661(代)

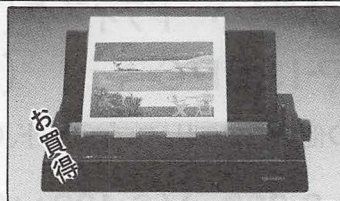
今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

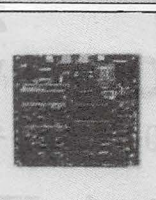
X68000の情報のすべて!(当店はX68000の認定代理店です。お気軽にご相談下さい)

CZ-8PC2 標準価格¥69,800 (今、プリンターを探してる人へ フタバのお買得情報。)

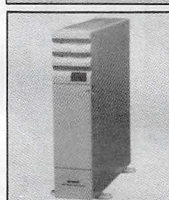
C.G.やカラーイメージボードで取り込んだ映像を7色で鮮やかにハードコピー。JIS第1水準・第2水準の漢字をはじめ、多彩な文字種が使え、文書作成には24×24ドットの高品位で対応。
●オプション黒色リボンカセット CZ-8PC-1 ¥700
カラーリボンカセット CZ-8PC-2 ¥800
※信号ケーブルおよび黒色/カラーリボンカセット各1個付

●CZ-8PC2の主な仕様

項目	仕様
印字方式	ドットマトリクス・ノンインパクト熱転写
文字種類	288種(英数・記号・カナ(またはひらがな)・その他)
文字ドット構成	19(タテ)×15(ヨコ)ドット(パイカ文字、エリート文字)、19(タテ)×10(ヨコ)ドット(縮小文字) 24(タテ)×24(ヨコ)ドット(漢字)、24(タテ)×12(ヨコ)ドット(半角文字)
印字速度	45字/秒(パイカ文字)、30字/秒(漢字)
電源・消費電力	AC100V、50/60Hz、40W(印字動作中)、15W(待機中)
外形寸法・重量	幅365×奥行253×高さ65(mm)、約3.5kg

特価¥4?,800
お支払例 ¥8,168× 6回 ¥5,038×10回
¥4,236×12回 ¥3,450×15回

CZ-6BC1

FAXボード。拡張I/Oスロットに装着し電話回線を利用してデーター通信を行う事ができる。

標準価格····¥79,800
特価 ¥6?,000
お支払例 ¥5,920×12回 ¥4,071×18回
¥3,712×20回 ¥3,147×24回

CZ-6TU

RGBシステムチューナカラーディスプレイで、テレビ番組が楽しめます(200ラインアナログRGB)、ビデオ入力端子付。

¥35,800
特価 ¥28,800
現 金 一 括 払

FDD搭載タイプの場合

実装密度をさらに追求、信頼性を高めたハイコストパフォーマンスFDモデル。HDモデル。

CZ-601C	¥319,800
CZ-611D	¥145,000
合 計	¥464,800

CZ-601C	¥319,800
CZ-601D	¥119,800
合 計	¥439,600

CZ-601C	¥319,800
CZ-603D	¥ 84,800
合 計	¥404,600

(こんな表示で申し訳ない! はっきり書くとおこられます。)

(注)CZ-611Dは0.31mmと、グラフィックをされる方にはすぐれたモニターです。CZ-601DはTVも写る、皆様に人気のあるモニターです。CZ-603DはTVは写りません、パソコンのディスプレイとしてご使用下さい。

※写真はCZ-601C-BK+CZ-603D-BK

HDD搭載タイプの場合

CZ-611C	¥399,800
CZ-611D	¥145,000
合 計	¥544,800

CZ-611C	¥399,800
CZ-601D	¥119,800
合 計	¥519,600

CZ-611C	¥399,800
CZ-603D	¥ 84,800
合 計	¥484,600

(申し訳ないが値段は言えない程安いんです。)
分割の価格はどうぞお気軽にお問合せ下さい。

初めて買われる方へ。
HDD搭載タイプと云うのは20MBのハードディスクが付いています。
とりあえず初心者には必要ないだろうと思われますので601C+601Dをお勧めします。又他のオプションと併売する場合は充分な値引も考慮します。

もちろん、分割払いもできます。

■本体+キーボード+マウス・トラックボール CZ-601C-GY(グレー)・-BK(ブラック)標準価格319,800円
■14型カラーディスプレイCZ-603D-GY(グレー)・-BK(ブラック)標準価格84,800円(チルトスタンド同梱)

型番	品名	標準価格	販売価格	お支払例
CU-14BD	ディスプレイ	¥ 64,800	¥ ?7,800	¥3,601×15回
CU-14ED	ディスプレイ	¥ 79,800	¥ ?2,600	¥3,346×18回
CU-14AD	ディスプレイ	¥ 84,800	¥ ?3,800	¥3,422×18回
CU-21CD	ディスプレイ	¥139,800	¥1?0,000	¥5,408×24回
CZ-820D	ディスプレイ	¥ 79,800	¥ ?4,800	¥3,375×15回
CZ-880D	ディスプレイ	¥109,800	¥ 8?,000	¥4,081×24回
CZ-603D	ディスプレイ	¥ 84,800	¥ ?3,800	¥3,137×24回
CZ-601D	ディスプレイ	¥119,800	¥ ??,000	¥3,203×36回
CZ-611D	ディスプレイ	¥145,000	¥1?3,000	¥3,892×36回
BF-68PRO	CRTフィルター	¥ 19,800	¥ 1?,800	現金一括払
CZ-502F	FDD(2DD)	¥ 99,800	¥ 7?,000	¥3,172×30回
CZ-503F	FDD(2D)	¥ 49,800	¥ 3?,800	¥3,219×12回
CZ-6BE1A	1MB (増設RAMボード)	¥ 38,000	¥ 3?,800	¥3,388×10回
CZ-6BE2	2MB (増設RAMボード)	¥ 79,800	¥ 6?,000	¥4,071×18回
CZ-6BE4	4MB (増設RAMボード)	¥138,000	¥1?8,000	¥3,720×36回

型番	品名	標準価格	販売価格	お支払例
AN-8TU	RGBシステムチューナ	¥ 35,800	¥ 28,800	現金一括払
CZ-8PK7	プリンタ(80桁)	¥122,000	¥ 9?,000	¥3,238×36回
CZ-8PK8	プリンタ(136桁)	¥152,000	¥1?7,000	¥3,169×48回
CZ-8PK9	プリンタ(80桁)	¥ 89,800	¥ ?0,000	¥3,442×24回
CZ-6VT1	カラーイメージユニット	¥ 69,800	¥ 5?,000	¥3,562×18回
CZ-8BV2	カラーイメージボード	¥ 39,800	¥ 31,800	¥3,498×10回
AN-160SP	アンプ内蔵スピーカー	¥ 59,800	¥ ?8,000	¥3,053×18回
CZ-8BS1	FM音源ボード	¥ 23,800	¥ 19,800	現金一括払
CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	¥ 24,000	現金一括払
CZ-6BU1	ユニバーサルIOボード	¥ 39,800	¥ 3?,000	¥3,520×10回
CZ-6BG1	GP-IBボード	¥ 59,800	¥ 4?,000	¥3,053×18回
CZ-8TM1	モデム	¥ 29,800	¥ 25,000	現金一括払
CZ-8TM2	モデム	¥ 49,800	¥ ?9,000	¥3,608×12回
CZ-8NT1	トラックボール	¥ 13,800	¥ 12,500	現金一括払
CZ-6SD1	システムラック	¥ 44,800	¥ 3?,800	¥3,312×12回

型番	品名	標準価格	販売価格	お支払例
CZ-6BF1	増設RS232Cボード	¥ 49,800	¥ 4?,000	¥3,013×15回
CZ-6BP1	数値プロセッサボード	¥ 79,800	¥ 6?,000	¥3,147×24回
CZ-6EB1	I/Oボックス	¥ 88,000	¥ 7?,000	¥3,442×24回
【OS9/X68K近日発売——予約販売開始! 先進のパソコンに応える先進のオペレーティングシステム】				
CZ-227BS	TOP財務会計	¥200,000	¥1?8,000	¥4,279×48回
CZ-213MS	MUSIC PRO-68K	¥ 18,800	¥ 15,800	現金一括払
CZ-214MS	SOUND PRO-68K	¥ 15,800	¥ 13,800	現金一括払
CZ-212BS	ビジネス PRO-68K	¥ 68,000	¥ 5?,000	¥3,435×18回
CZ-211LS	Cコンパイラ PRO-68K	¥ 39,800	¥ 3?,000	¥3,520×10回
CZ-141SF	NEW-Z BASIC	¥ 18,800	¥ 15,800	現金一括払
CZ-137SF	turbo Z's STAFF	¥ 19,800	¥ 16,800	現金一括払
CZ-133SF	モデムターミナルソフト	¥ 25,800	¥ 2?,000	現金一括払
	Z'STAFF PRO-68K	¥ 58,000	¥ 47,000	¥3,541×15回
	kamikaze	¥ 68,000	¥ 5?,000	¥3,499×18回

●分割回数は3回〜48回まで自由に選べます。

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

X1turboZ

NEW-Z BASICは後で買えばいい。ハイグレードモニタをセットして驚異の価格。

CZ-880C····¥218,000
CZ-880D····¥109,800
合計·········¥327,800

特価 ¥1?3,000
お支払例 ¥16,003×12回 ¥8,506×24回
¥5,959×36回 ¥4,685×48回

X1turboZII

X1turboZの本格派セット。TV付2モードオートスキャンディスプレイ。

CZ-881C····¥179,800
CZ-880D····¥109,800
合計·········¥289,600

特価 ¥2?4,000
お支払例 ¥20,720×12回 ¥11,013×24回
¥7,716×36回 ¥6,067×48回

X1twin

HEシステムを搭載、最上級ゲーム機とパソコンが合体。

CZ-830C····¥ 99,800
CZ-820C····¥ 79,800
合計·········¥179,600

特価 ¥94,800
お支払例 ¥8,769×12回 ¥6,030×18回
¥4,661×24回 ¥3,265×36回

X1Gmodel30

X1Gの本格派セットFDD2基内蔵、専用カラーモニタはTVにも使用可能。

CZ-822C····¥118,000
CZ-820D····¥ 79,000
合計·········¥197,000

特価 ¥79,800
お支払例 ¥7,382×12回 ¥5,076×18回
¥3,924×24回 ¥3,245×30回

●頭金なし(手軽な電話クレジット) ●製品先取り(お支払いは約1〜2ヶ月後から) ●低金利クレジット(1回の支払いは2,700円以上で3〜48回。ボーナス併用も可) ●カレッジクレジット(保証人なし。但し満20歳以上の学生の方) ●18歳未満の方(ご両親が代理購入者としてお申し込み下さい) ●納期(通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れることがありますので御了承下さい) ●完全保証(すべてメーカー保証書付。アフターケア万全) ●全国代引(お届けした者に、代金をお支払いいただく方法です。但し手数料1,000円)

AM10時からPM9時まで受付 日曜・祝日も営業

パソコンプラザ

店頭セール実施中

'88オクトで始まるパソコンワールド

☎ 03-730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00
〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販 電話一本で、ハイ即納 ●年中無休

OCT-1 システム インフォメーション
- ▶全商品保証付(メーカー保証＋当社保障1年)
- ▶超低金利ハッピークレジット(1回～60回)頭金ナシOK!
- ▶冬のボーナス一括払いOK! ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト
セレクテッドシステム

'88オクトで始まるパソコンワールド。ウィンターセール開催中!!

12/1～1/15の期間中、店頭でお買い上げの方に、ゲームソフト25%OFF!!

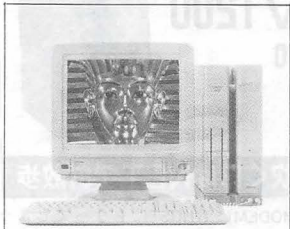
蒲田 年忘れ! 年末・年始大感謝祭セール

OPEN

X68000フェア開催中!!

《プレゼント実施中》★ドラゴンスピリッツ(¥8,800)★X-68000ポーチ ★3MブラックディスクMD-2HD10枚)

お好みのセットをお選び下さい。
送料無料

20MBハードディスク・モデル

X68000 ACE-HD

CZ-611C-GY/BK
定価¥399,800
現金特価!!
お電話下さい。
推 選

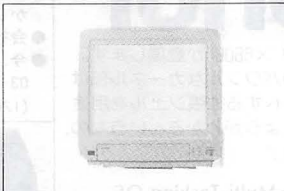
ハイコストパフォーマンス
FDモデル

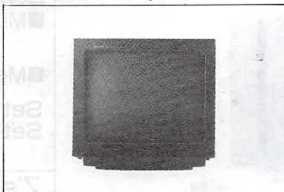
X68000 ACE

CZ-601C-GY/BK
定価¥319,800
現金特価!!
お電話下さい。
推 選

15型カラーディスプレイTV

CZ-611D-GY/BK
定価¥145,000

15型カラーディスプレイTV

CZ-601D-GY/BK
定価¥119,800

14型カラーディスプレー

CZ-603D-GY/BK
定価¥84,800

21型カラーディスプレイ

CU-21CD
定価¥139,800

ACE-HD
①CZ-611C＋CZ-611D
合計金額¥544,800……………大特価TEL下さい。

12回	¥36,200	24回	¥18,900	36回	¥13,000	42回	¥11,400

ACE
②CZ-601C＋CZ611D
合計金額¥464,800……………大特価TEL下さい。

12回	¥29,800	24回	¥15,600	36回	¥10,700	42回	¥9,400

ACE-HD
③CZ-611C＋CZ-601D
合計金額¥519,600……………大特価TEL下さい。

12回	¥34,400	24回	¥18,000	36回	¥12,400	42回	¥10,500

ACE
④CZ-601C＋CZ-601D
合計金額¥439,600……………大特価TEL下さい。

12回	¥28,000	24回	¥14,700	36回	¥10,150	42回	¥8,850

ACE-HD
⑤CZ-611C＋CZ-603D
合計金額¥484,600……………大特価TEL下さい。

12回	¥31,800	24回	¥16,600	36回	¥11,500	42回	¥15,000

ACE
⑥CZ-601C＋CZ-603D
合計金額¥404,600……………大特価TEL下さい。

12回	¥25,700	24回	¥13,500	36回	¥9,300	42回	¥8,100

ACE-HD
⑦CZ-611C＋CU-21CD
合計金額¥539,600……………超特価TEL下さい。

12回	¥35,100	24回	¥18,400	36回	¥12,700	42回	¥11,000

ACE
⑧CZ-601C＋CU-21CD
合計金額¥459,600……………超特価TEL下さい。

12回	¥29,100	24回	¥15,200	36回	¥10,500	42回	¥9,100

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料¥2,000
※掲載の価格は11/20現在ですので、まずは、お電話にてご確認ください。

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

※X68000セットお買上げの方には、ドラゴンスピリット(¥8,000)をプレゼント!!

広告掲載商品以外の製品も取扱っております。

X68000周辺機器大セール実施中　全商品送料無料

パソコンチューナー

AN-8TV(定価35,800)

- CU-21CDでTVが見れる!!
- ビデオ入力端子
- モニター出力端子
- スーパーインポーズ表示可能

大特価¥ 88,800

カラーイメージスキャナ

CZ-8NS1(定価¥188,000)

- A-4サイズまでの写真・図形フルカラー読み取り
- 縮少・拡大自在

大特価¥145,000

カラービデオプリンタ

CZ-6PV1(定価¥198,000)

- オリジナルCGや取り込んだ画像を色鮮やかにコピー!!

大特価¥155,000

カラーイメージユニット

CZ-6VT1-BK(定価¥69,800)

- イメージ豊かなアートワークをサポート!! おしゃれなアートが楽しめます。

大特価¥ 55,000

型　名	商　品	定　価	特　価
CZ-6BELA	IBM増設RAMボード	¥38,000	¥30,000
CZ-6BUI	ユニバーサルI/Oボード	¥39,800	¥31,800
CZ-6BGI	GP-IBボード	¥59,800	¥47,000
CZ-6BPI	プロセッサ・ボード	¥79,800	¥63,000
CZ-6BCI	FAXボード	¥79,800	¥63,000
CZ-6BMI	MIDボード	¥26,800	¥21,000

型　名	商　品	安　価	特　価
CZ-6EBI	拡張I/Oボックス	¥88,000	¥70,000
CZ-8TMZ	モデムユニット	¥49,800	¥39,800
CZ-6BN-I	スキャナ用パラレルボード	¥29,800	¥23,000
CZ-8NTI	トラックボール	¥13,800	¥11,000
AN-160SP	アンプ内蔵スピーカー	¥59,800	¥47,800

熱転写カラー漢字プリント　プリンター

CZ-8PC3 ¥65,800

- 24ドットサーマルヘッド
- B5～B4縦まで
- ハガキ可能
- カラーハードコピー可能

大特価¥51,000

①CZ-8PK7(24ピン80桁)
定価¥122,000
大特価・TEL下さい。

②CZ-8PK8 (24ピン136桁)
定価¥152,000
大特価・TEL下さい。

③CZ-8PK9
定価¥89,800
大特価・TEL下さい。

パソコンラック(4段)

推奨

キミだけのパソコン・スペースを作っちゃおう!!
移動自由自在
サイズ
1245(H)×614(D)×600(W) m/m
定価¥22,800
大特価¥12,000

X68000ソフト大セール実施中　※ゲームソフトオール23%off

〈グラフィック〉
● Z's STAFF PRO68
(シャフト)
定価¥58,000
大特価 ¥41,000

〈データベース〉
● KAMIKAZE
(サムシンググッド)
定価¥68,000
大特価 ¥47,000

〈グラフィック〉
● C-TRACE68
(キャスト)
定価¥68,000
大特価 ¥51,000

型　名	商　品	定　価	特　価
BUSINESS PRO68K	統合型表計算	¥68,000	¥55,000
CARD PRO68K	カード型データベース	¥29,800	¥23,800
DATA	コマンド型データベース	¥58,000	¥45,000
COMMUNICATION PRO68K	通信ソフト	¥19,800	¥15,000
SAMPLING PRO68K	サンプリングエディタ	¥17,800	¥14,000
MUSIC PRO68K	楽譜ワープロ	¥18,800	¥15,000
SOUND PRO68K	サウンドエディタ	¥15,800	¥13,000
NEW PRINT SHOP PRO68K	ポップアートツール	¥19,800	¥15,000
C-COMPILER PRO68K	Cコンパイラ	¥39,800	¥32,000
EW	ワープロ	¥38,000	¥29,800
G-68	グラフィックツール	¥14,800	¥12,000
E-68K	スプライトエディタ	¥19,800	¥16,000

店頭ゲームソフトオール23%off! ビジネスソフト20%より特価中

★通信販売お申込みのご案内★

〒144 東京都大田区蒲田4-6-7　TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

振込先
三菱銀行 蒲田支店 ㊚No.0278691
東京都民銀行 蒲田支店 ㊚No.0320955
株式会社 億人(オクト)

▶ソフトのご注文は、OCT-2にて取扱っております

※掲載の価格は11/20現在ですので、まずは、お電話にてご確認ください。

'88オクトで始まるパソコンワールド。ウィンターセール開催中!!

SHARP
パーソナルコンピュータMZ-2861

これ一台で、アレコレできます。

高性能ワープロ+高性能パソコン

- ●日本語ワープロ「書院28」搭載！
- ●MS-DOS™V3.1標準装備！

MZ-2531 V.2 BASIC +通信ソフト プレゼント
'88・12/1日〜'89・2/28日

16ビットパーソナルコンピュータ

MZ-2861

MZ-2861/328,000円＋
モニターMZ-1D10/41,800円
合計369,800円

43% OFF

超特価!!! 210,000円

下取りセールもOKです。

タイプ、消耗程度により査定致しますので、詳しくは電話でお問い合わせください。(0426-45-3001〜3)

一例…MZ-2521下取りの場合、超特価165,000円

OAソフトウェアも大充実！

UPシリーズ

MZ-2861の日本語入力機能を有機的に活かす統合OAソフトウェア「UPシリーズ」。ディスクパブリッシングという新しいジャンルのレイアウトワープロ、集計表・グラフ作成統合ソフトウェア、自由度の高いカード型データベース、アウトラインプロセッサというジャンルの新しい企画書作成ソフトウェア。オフィスワークを代表的な4つの局面からアプローチして専門化した、切れモノOAツールです。

- 日本語レイアウトワープロ■デスクUP(1P-1251) 定価¥88,000➡特価¥70,400
- 集計表・グラフ作成ソフト■チャートUP(1P-1252) 定価¥55,000➡特価¥44,000
- カード型データベース■UPクリッパー(1P-1253) 定価¥77,000➡特価¥61,600
- 企画書作成ソフト■プランUP(1P-1254) 定価¥66,000➡特価¥52,800

MZ-2861の多彩な周辺機器

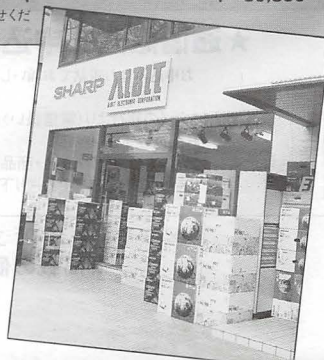
- ●MZ-1D26(14型カラーディスプレイ)……¥ 89,800
- ●MZ-1D27(15型カラーディスプレイTV付)……¥127,000
- ●Cu-14BD(14型カラーディスプレイAN1508付)……¥ 66,000
- ●SSSC28……¥ 79,600
- ●SSSC28M(モノクロハンディスキャナ)……¥ 49,800
- ●MZ-1P23(レーザープリンター)……¥950,000
- ●MZ-1P27(水平インサートプリンタ)……¥268,000
- ●MZ-1P28(80桁漢字プリンタ)……¥148,000
- ●MZ-1P29(136桁漢字プリンタ)……¥168,000
- ●IO-730(136桁インクジェットプリンタ)……¥230,000
- ●MZ-1R35(1MBRAM)……¥ 55,000
- ●MZ-1R36(1MB増設RAM)……¥ 45,000
- ●MZ-1V01(イメージ情報ステーション)……¥278,000
- ●IP-1256(パソコンFAX28)……¥ 99,800
- ●MZ-1X29(マウス)……¥ 13,800
- ●MZ-1X30(モデムホン)……¥ 98,000
- ●MZ-1F23(20MBハードディスク)……¥ 29,800
- ●MZ-1E35(ADPCMボード)……¥ 49,800
- ●MZ-1E39(RS232C/2ch)ボード……¥ 39,800

※表示の価格は定価につき、割引価格はお問い合せください。

☎0426-45-3001〜3
FAX.0426-44-6002

●営業時間/10:00〜19:00●電話受付/20:00迄可●定休日/日曜日(祭日営業)

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

X1·MZ周辺機器他、シャープ製品徹底の品揃え

もちろん本体新製品から他店では入手しにくい旧タイプ周辺機器まで全品新品保証付。しかも大特価徹底の品揃え。特にひとつ前のタイプは絶対のお買い得です。(旧タイプは限定数のため、電話で在庫をお確かめの上ご注文ください。)

AIBIT
アイビット電子株式会社

特別セット(本体+ディスプレイ)

●シャープX1 Gmodel 30セット
(CZ822CB(本体)
CZ820DB(ディスプレイ))
大特価¥79,800

●シャープX1 twinセット
(CZ830CBK(本体)
CZ820DB(ディスプレイ))
大特価¥94,800

特別セットX68000 在庫処分セール!

5年先を見つめたコンセプトマシン。このマシンのポテンシャルにふさわしい数々のソフトウェアーの登場で新たな局面。絶対お買い得!です。

●セット内容
本体/X68000(CZ-600C)¥369,000
ディスプレイ/CZ-603D ¥84,800
定価合計¥453,800➡超特価¥298,000
※新古品(メーカー保証付)セットもあります。
同セット内容で合計¥279,000

X68000通信ソフトセット
●CZ-223CS ¥19,800
●CZ-8TM1 ¥29,800
定価合計¥49,600➡¥19,800

CZ-218AS入荷しました。X68000用サラマンダー
¥8,800⇨¥7,000

アイビット推奨ディスプレイ

●シャープCu21CD(21型)
マルチスキャン方式
(アナログ)
定価¥139,800⇨特価
特価¥110,000

CU21CD対応パソコン機種:CZ880C/881C/600C/611C。PC88VA/VA2/VA3/MK2SR/TR/FR/MR。PC8801FH/MH/FA/MA。PC286U/V/L。PC9801U/UV/UX/VM/VX/LV各シリーズ。ケーブルは付属を使用(X1シリーズはAN1506で使用)MZ700/1500/2000/2200/2500はAN1508で。

●シャープCZ-880D-GY
(14型)TV付(2000/4000)
(デジタル/アナログ)
定価¥109,800⇨
特価¥69,800

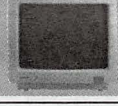
CZ880DGY対応パソコン機種:CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。PC88VA/VA2/VA3/MK2SR/TR/FR/MR。PC9801U/UV/UX/VM/VX/LV各シリーズ。アナログ25ピン↔25ピンケーブルを使用(デジタルは各専用ケーブルで)。MZ700/1500/2000/2200/2500各シリーズ(推奨品シャープ8D8K)。

●シャープCZ-820D
(14型)TV付
(2000デジタル)
定価¥79,800⇨
特価¥39,800

CZ820D対応パソコン機種:CZ880C/881C。X1/TURBOシリーズ(X1モードのみ)ケーブルは付属を使用。MZ700/1500/2000/2200シリーズ(推奨品シャープ8D8K)。その他デジタル表示は各専用ケーブルで。

●シャープCu-15M1
(15型デジタル/アナログ)
定価¥99,800⇨
特価¥79,800

Cu-15M1対応パソコン機種:CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。PC88VA/VA2/VA3/MK2SR/TR/FR/MR。PC9801U/UV/UX/VM/VX/LV各シリーズ。アナログ25ピン↔25ピンケーブルを使用(デジタルは各専用ケーブルで)。MZ700/1500/2000/2200/2500各シリーズ(推奨品シャープ8D8K)。

●シャープMZ-1D24
(14型)(4000アナログ8ピン)
定価¥128,000⇨
特価¥96,000

MZ-1D26対応パソコン機種:MZ2500/2800シリーズ専用。

●シャープCu14ED
(14型)(2000/4000)
定価¥79,800⇨
特価¥54,800

Cu14ED対応パソコン機種:CZ880C/881C(X1/TURBOシリーズはAN506使用)。PC88/VA2/VA3/MkIISR/MR/FR/TR。PC8801FH/MH/FA/MA。PC286U/V/L。PC9801UV/UX/VM/VX/LV各シリーズは付属ケーブルを使用。

●シャープCu14BD
(14型)(2000/4000)
(アナログ)
定価¥64,800⇨
特価¥49,800

Cu14BD対応パソコン機種:CZ880C/881C(X1/TURBOシリーズはAN506使用)。PC88/VA2/VA3/MkIISR/MR/FR/TR。PC8801FH/MH/FA/MA。PC286U/V/L。PC9801UV/UX/VM/VX/LV各シリーズは付属ケーブルを使用。

●富士通ゼネラルDM405
(14型)
(2000アナログ21/8ピン)
定価¥67,800⇨
特価¥36,000

DM405対応パソコン機種:MSX2。X1シリーズ。MZ700/1500/2000/2200シリーズ。FM77AV/7/8シリーズ。(ケーブルは各専用のものを使用)

●富士通FM-TV151(15型)
TV付カラー
定価¥89,800⇨
特価¥48,000

FM-TV151対応パソコン機種:MSX2。X1シリーズ。MZ700/1500/2000/2200シリーズ。FM77AV/7/8シリーズ。(ケーブルは各専用のものを使用。)

本 体

●シャープCZ-820C …………¥69,800⇨¥16,800
●シャープCZ-601CX68000ACE ……¥319,800⇨超特価☎
●シャープCZ-611CX68000ACE HD‥¥399,800⇨超特価☎
●シャープCZ-822C ……………………¥59,800
●シャープCZ-880C-BK(X1 turbo ZIII) 新発売
●シャープCZ-880C ………¥218,000⇨ ¥95,000
●シャープMZ-2861+1P-1252‥¥383,000⇨¥245,000
●シャープMZ-5521…………¥388,000⇨¥65,000
●シャープMZ-2520…………¥159,800⇨¥78,000
●シャープMZ-2521…………¥198,000⇨¥85,000
●NEC PC-9801VX4 ………¥643,000⇨¥360,000
●NEC PC-9801XA2………¥695,000⇨¥149,000
●NEC PC-98LT11…………¥238,000⇨¥119,800
●NEC PC-98LT21…………¥288,000⇨¥149,800
●富士通FM-AV771…………¥128,000⇨¥45,000
●富士通FM-AV772…………¥158,000⇨¥55,000
●富士通AM-AV40…………¥228,000⇨¥95,000
●富士通16βFD……………¥400,000⇨¥180,000
●富士通16βキーボード………¥25,000⇨¥20,000

拡張機器他

●シャープCZ-8TM1 …………¥29,800⇨¥9,800
●シャープMZ-1E29 …………¥17,800⇨¥9,800
●シャープX1用ジョイカード……………………¥1,500
●シャープCZ-8EB-3 …………¥33,800⇨¥28,000
●シャープCZ-8EP …………¥11,800⇨¥9,000
●シャープMZ-1U05…………¥12,000⇨¥8,500
●シャープMZ-1U09 …………¥9,000⇨¥7,200
●シャープMZ-1E24232Cカード…¥19,800⇨¥16,800
●シャープCZ-8BK3…………¥13,800⇨¥11,700
●シャープCZ-8BK4…………¥6,800⇨¥5,700
●シャープMZ-1M03…………¥69,000⇨¥35,000
●シャープMZ8BC04…………¥18,000⇨¥8,000
●シャープMZ-8B104 …………¥45,000⇨¥18,000
●シャープMZ-1R09…………¥35,000⇨¥25,000
●シャープMZ-1R10…………¥30,000⇨¥12,000
●シャープMZ-1R11 …………¥80,000⇨¥40,000
●シャープMZ-1R19…………¥35,000⇨¥15,000
●シャープMZ-1R24 …………¥22,000⇨¥6,000
●シャープMZ-1R26A…………¥15,000⇨¥12,800
●シャープMZ-1R27A…………¥13,000⇨¥10,000
●シャープMZ-1R28A…………¥13,000⇨ 品切
●シャープMZ-1R29A…………¥32,000⇨¥10,000
●シャープMZ-1R37…………¥35,800⇨¥28,000
●シャープMZ-1T02 …………¥19,800⇨¥8,500
●シャープMZ-1T03 …………¥12,000⇨¥8,500
●シャープCZ-8BGR2…………¥14,800⇨¥4,000
●シャープCZ-8BS1…………¥23,800⇨¥19,500
●シャープCZ-51F同等品……………………¥22,000
●シャープCZ-52F同等品……………………¥20,000
●シャープMZ-2000/2200/80B/1500/700用
(フロッピーインターフェース)¥23,500⇨¥18,000
●シャープX1、MZ用マウス …………特価¥4,800
●シャープMZ-1X29…………¥13,800⇨¥11,000
●シャープMZ-1M08…………¥10,000⇨¥6,000
●シャープMZ-3500キーボード……………¥10,000
●シャープMZ-5500キーボード……………¥10,000
●シャープX1シリーズ用キーボード………¥10,000
●シャープMZ-2000/2200通信セット
MZ-1E29+MZ-1X22+MZ-2Z052‥‥¥49,100⇨¥20,000
●シャープMZ-1E26…………¥24,800⇨¥13,000

プリンター

●シャープMZ-1P27………¥268,000⇨¥214,400
●シャープMZ-1P28………¥148,000⇨¥118,400
●シャープMZ-1P29………¥168,000⇨¥134,400
●シャープMZ-1P17(ケーブルプリンター)¥85,800⇨¥39,800
●シャープMZ-6P11…………¥95,000⇨¥35,000
●シャープCZ-8PC2…………¥69,800⇨¥49,800
●シャープCZ-8PC3…………¥65,800⇨¥52,000
●シャープCZ-8PD2…………¥79,800⇨¥25,000
●シャープMZ-8PD3…………¥59,800⇨¥16,000
●シャープMZ-1P10A………¥245,000⇨¥80,000
●シャープCZ-8PK5…………¥129,000⇨¥59,800
●シャープCZ-8PK6…………¥159,000⇨¥69,800

フロッピーディスク

●シャープCZ-503F…………¥49,800⇨¥34,000
●シャープCZ-503F(インターフェースカードなし)……¥30,000
●シャープCZ-502F…………¥99,800⇨¥75,000
●シャープCZ-300F(CZ3PCM)………………¥13,000

ソフト

●シャープCZ-141SF…………¥18,800⇨¥16,000
●シャープMZ-2Z013…………¥25,000⇨¥21,000
●シャープMZ-2Z017…………¥20,000⇨¥17,000
●シャープMZ-2Z032…………¥12,000⇨¥6,000
●シャープMZ-2Z064…………¥69,800⇨¥59,500
●シャープMZ-2Z023…………¥50,000⇨¥42,500
●シャープMZ-2Z025…………¥49,800⇨¥15,000
●シャープMZ-2Z014…………¥68,000⇨¥15,000
●シャープMZ5Z013…………¥6,500⇨¥2,000
●シャープ6F03……………………10枚¥4,000
●シャープMZ-6Z010 …………¥10,000⇨¥8,500
●シャープMZ-1M01……………………特価¥8,500

X68000関係ソフト

●CZ-220BS……………………………¥46,400
●CZ-226BS……………………………¥24,000
●シャープCZ-21SMS(サンプリングPRO68K)
……………………¥17,800⇨¥14,200
●CZ-227BS ……………………………¥160,000
●シャープCZ-211LS…………¥39,800⇨¥35,800
●シャープCZ-6BE1 …………¥35,000⇨¥29,800
●シャープCZ-6BE1A …………¥38,000⇨¥32,300

富士通OS9関係ソフト

●FM-16β日本語MS-DOSB278A100‥¥32,000⇨¥25,600
●FM-16β日本語CP/M86V1.0B271A100‥¥25,000⇨¥19,500
●FM-7, 77/L2/L4OS-9LV,1SMO7317-M143‥¥48,000⇨¥39,400
●FM-77/L4OS-9LV,2SMO7317-M144……¥58,000⇨¥47,600
●FM-77AV OS-9LV,2B273A030…………¥30,000⇨¥24,600

SHARPポケットコンピュータ

●PC-1360…………………¥29,800⇨¥19,800
●PCE-200…………………¥22,000⇨¥17,800
●PCE-500…………………¥28,800⇨¥24,800
●CE-152……………………¥19,800⇨¥9,800
●CE-161プログラムモジュール ……¥50,000⇨2個¥7,000
●CE-159プログラムモジュール………¥35,000⇨¥4,200
●シャープCE-140PK………… ¥43,000⇨16,000

ポケコン総合カタログ並びに特価表を差し上げます。
切手¥70を同封の上、当社へお申込みください。

本誌発売時には、上記価格よりさらにお求めやすい価格に変更されている場合があります。

●MZ2500シリーズ用ソフト大量在庫(限定1000本大放出!)

☎0426-45-3001~3
FAX.0426-44-6002
●営業時間/10:00~19:00●電話受付/20:00迄可●定休日/日曜日(祭日営業)

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

信用をモットーに、よりよい品をより安く、迅速にお届けします。

全国通信販売
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 (普)1752505

粗品プレゼント! 今、お買い上げになるとシャープ特製フロッピーホルダーがもらえる!

超低金利クレジットOK!!

またまた秋葉原でおなじみの

12/15～1/20

- ●お近くの方はお
- ●本体単品で特
- ●ビジネスソフト定

ご利用下さい。

超低金利クレジット

回数	金利
12回	4.5%
24回	9.5%
36回	13%
48回	17%
60回	22%

X68000ACE HD (送料¥2,000)

NEW
CZ-603D
(定価¥84,800)
- ●0.31ピッチ
- ●14インチ
- ●TVチューナーなし

Ⓐセット：CZ-611C＋CZ-611D＋M-2HD(10枚)
………定価¥544,800➡P&A超特価(価格はお電話下さい)

12回	34,800	24回	18,200	36回	12,500	48回	9,700	60回	8,100

Ⓑセット：CZ-611C＋CZ-601D＋M-2HD(10枚)
………定価¥519,600➡P&A超特価(価格はお電話下さい)

12回	33,000	24回	17,300	36回	11,900	48回	9,300	60回	7,700

Ⓒセット：CZ-611C＋CZ-603D＋M-2HD(10枚)
………定価¥484,600➡P&A超特価

12回	30,000	24回	15,700	36回	10,800	48回	8,400	60回	7,000

※X68000セットでお買い上げの方にドラゴンスピリッツ¥8,800をプレゼント致します。
※チルトスタンド(CZ6ST1 ¥5,800)必要な方は¥5,000加算して下さい。

X68000ACE (送料¥2,000)

X68000用
ジョイスティック
送料¥500
- ●XE-1PRO
定価¥9500
特価¥7,800
- ●ASCII STICK
X-TURBO
定価¥6,800▶
特価¥5,500

Ⓐセット：CZ-601C＋CZ-611D＋M-2HD(10枚)
………定価¥464,800➡P&A超特価(価格はお電話下さい)

12回	29,100	24回	15,200	36回	10,500	48回	8,100	60回	6,800

Ⓑセット：CZ-601C＋601D＋M-2HD(10枚)
………定価¥439,600➡P&A超特価(価格はお電話下さい)

12回	27,500	24回	14,400	36回	9,900	48回	7,700	60回	6,400

Ⓒセット：CZ-601C＋CZ-603D＋M-2HD(10枚)
………定価¥404,600➡P&A超特価(価格はお電話下さい)

12回	24,300	24回	12,800	36回	8,800	48回	6,800	60回	5,700

※チルトスタンド(CZ-6STI ¥5,800)必要な方は¥5,000加算して下さい。
※X 68000セットでお買い上げの方にドラゴンスピリッツ¥8,800をプレゼント致します。

X-1ターボZⅢ/ZⅡ/Z (セットでお買い上げの方にディスケット10枚 ジョイカード、ゲーム3種プレゼント) 送料¥2,000

Ⓐセット：NEW
X-1ターボZⅢ(CZ-888C＋CZ-860D)
定価¥269,600▶価格はお電話下さい

12回	17,200	24回	9,000	36回	6,200	48回	4,800	60回	4,000

Ⓑセット：NEW
X-1ターボZⅢ(CZ-888C＋CZ-830D)
定価¥269,600▶価格はお電話下さい

12回	16,400	24回	8,600	36回	5,900	48回	4,600	60回	3,800

Ⓒセット：X-1ターボZⅡ
(CZ-881C＋CZ-880D)
定価¥289,600▶特価**¥182,000**

Ⓓセット：X-1ターボZ
(CZ-880C＋CZ-880D)
定価¥327,800▶特価**¥158,000**

X-1 TWIN/G (送料¥2,000)

Ⓐセット：X-1TWIN(CZ-830C＋CZ-820D)…………定価¥179,600▶特価**¥94,000**
Ⓑセット：X-1Gモデル30(CZ-822C＋CZ-820D)……定価¥197,800▶特価**¥79,000**
●セットでお買い上げの方に、ディスケット10枚、ジョイカード、ゲーム3種、パソコンラックⒶ3段をプレゼント

プリンターセット ※全セットにケーブル、用紙付 (送料¥1,000)

Ⓐセット：CZ-8PC2 限定……………定価¥69,800▶特価**¥44,000**
Ⓑセット：CZ-8PC3……………定価¥65,800▶特価**¥51,000**
Ⓒセット：CZ-8PK7…………定価¥122,000▶P&A超特価(お電話下さい)

12回	8,100	24回	4,200	36回	3,500

Ⓓセット：CZ-8PK8………定価¥152,000▶P&A超特価(お電話下さい)

12回	11,000	24回	5,300	36回	3,600

Ⓔセット：CZ-8PK9………定価¥89,800▶P&A超特価(お電話下さい)

12回	6,000	24回	3,100

Ⓕセット：CZ-8PK6……………………定価¥159,000▶超特価**¥69,000**
限定品、用紙1,000枚付、送料無料

お知らせ 11月1日より、P&A本店の住所が変わります。装いも新たに大きくなりました。今後とも、ご愛顧下さるようお願い致します。

回～60回払いまでOK!!

★頭金なし! ★即日発送

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

X68000用ソフトコーナー(送料¥1,000)

	品名	定価	特価
Ⓐ	CZ-212BS(BUSINESS)	定価¥68,000	特価¥55,000
Ⓑ	CZ-220SB(DATA)	定価¥58,000	特価¥45,000
Ⓒ	CZ-226BS(CARD)	定価¥29,800	特価¥24,000
Ⓓ	CZ-213MS(MUSIC)	定価¥18,800	特価¥15,000
Ⓔ	CZ-214MS(SOUND)	定価¥15,800	特価¥12,500
Ⓕ	CZ-215MS(Sampling)	定価¥17,800	特価¥14,000
Ⓖ	CZ-221HS(NEW Print shop)	定価¥19,800	特価¥16,000
Ⓗ	CZ-223CS(Communication)	定価¥19,800	特価¥16,000
Ⓘ	CZ-211LS(C. compiler)	定価¥39,800	特価¥32,000
Ⓙ	CZ-224LS(福袋)	定価¥9,800	特価¥8,000
Ⓚ	Z's STAFF PRO-68K(シャフト)	定価¥58,000	特価¥42,000
Ⓛ	神風(サムシンググッド)	定価¥68,000	特価¥49,000
Ⓜ	ビジネスAD68K(マッショシステム)	定価¥98,000	特価¥78,500
Ⓝ	弥生(日本マイコン)	定価¥80,000	特価¥64,000
Ⓞ	CP/M-68K(ニューウェイブ)	定価¥110,000	特価¥88,000
Ⓟ	EW&EI(イースト)	定価¥38,000	特価¥30,500
Ⓠ	C-TRACE(キャスト)	定価¥68,000	特価¥54,500
Ⓡ	SHOGUN(サムシンググッド)	定価¥34,800	特価¥25,000
Ⓢ	SAMURAI(サムシンググッド)	定価¥19,800	特価¥15,200

カラービデオプリンター (送料¥1,000)

Ⓐセット:CZ-6PVI 定価¥198,000➡超特価¥155,000

12回	13,400	24回	7,000	36回	4,800	48回	3,700

カラーイメージスキャナ (送料¥1,000)

Ⓐセット:CZ-8NSI 定価¥188,000➡超特価¥145,000

12回	12,600	24回	6,600	36回	4,500	48回	3,500

周辺機器コーナー(送料¥1,000)●その他の周辺機器はお電話下さい。

	品名	定価	特価
Ⓐ	CZ-8BSI(FM音源ボード)	定価¥23,800	特価¥19,000
Ⓑ	CZ-8RLI(データレコーダ)	定価¥24,800	特価¥20,000
Ⓒ	CZ-8AV2(カラーイメージボードⅡ)	定価¥39,800	特価¥31,000
Ⓓ	CZ-8BRI(立体映像セット)	定価¥29,800	特価¥23,000
Ⓔ	CZ-8DT2(パーソナルテロッパ)	定価¥44,800	特価¥35,000
Ⓕ	CZ-6VTI(カラーイメージユニット)	定価¥69,800	特価¥55,000
Ⓖ	CZ-6EBI(拡張I/Oボックス)	定価¥88,000	特価¥69,000
Ⓗ	AN-160SP(アンプ内蔵スピーカーシステム)	定価¥59,800	特価¥47,000

ゲームソフト(1ヶ～20ヶまで送料¥500)

機種		品名	定価	特価
X68000用	Ⓐ	源平討魔伝(電波新聞社)	定価¥7,800	特価¥6,200
	Ⓑ	ドラゴンスピリット(電波新聞社)	定価¥8,800	特価¥7,000
	Ⓒ	スペースハリアー(電波新聞社)	定価¥6,800	特価¥5,400
	Ⓓ	熱血高校ドッジボール部(SHARP)	定価¥7,800	特価¥6,200
	Ⓔ	沙羅曼蛇(SHARP)	定価¥8,800	特価¥7,000
	Ⓕ	フルスロットル(SHARP)	定価¥8,800	特価¥7,000
	Ⓖ	琥珀色の遺言(リバーヒルソフト)	定価¥9,800	特価¥7,800
	Ⓗ	ザ・スーパーラスベガス(日本デグスタ)	定価¥12,800	特価¥10,200
	Ⓘ	マイト・アンド・マジック(スタークラフト)	定価¥9,800	特価¥7,800
	Ⓙ	ザ・リターン・オブ・イシター(SPS)	定価¥7,800	特価¥6,200
	Ⓚ	信長の野望(全国版)(KOEI)	定価¥9,800	特価¥7,800
	Ⓛ	麻雀悟空(シャノアール)	定価¥7,800	特価¥6,200
X-1ターボ用	Ⓐ	イースII(日本ファルコム)	定価¥7,800	特価¥6,800
	Ⓑ	ラスト・ハルマゲドン(ブレイングレイ)	定価¥7,800	特価¥6,200
	Ⓒ	ソーサリアン(日本ファルコム)	定価¥9,800	特価¥7,800
	Ⓓ	ハイドライド3(T&E SOFT)	定価¥7,800	特価¥6,200
	Ⓔ	アークス(ウルフチーム)	定価¥9,800	特価¥7,800
	Ⓕ	マスター・オブ・モンスター(システムソフト)	定価¥8,800	特価¥6,400
	Ⓖ	エグザイル(日本テレネット)	定価¥8,800	特価¥7,000
	Ⓗ	白夜物語(イーストキューブ)	定価¥7,800	特価¥6,200

P&A 特選パソコンラック (送料無料)

Ⓐ3段 875(H)×580(D)×610(W) ¥9,000
Ⓑ4段 1320(H)×600(D)×630(W) ¥12,500
Ⓒ5段 1280(H)×600(D)×620(W) ¥16,500

周辺機器コーナー (送料¥1,000)

	品名	定価	特価
Ⓐ	CZ-8BV2(カラーイメージボードⅡ)	定価¥39,800	特価¥31,000
Ⓑ	CZ-6VT1(カラーイメージユニット)	定価¥69,800	特価¥54,000
Ⓒ	CZ-6EB1(拡張I/Oボックス)	定価¥88,000	特価¥67,500
Ⓓ	AN-160SP(スピーカーシステム)	定価¥59,800	特価¥46,000
Ⓔ	CZ-6BE1A(1M RAM)	定価¥38,000	特価¥29,000
Ⓕ	CZ-6BP1(数値洋算)	定価¥79,800	特価¥61,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884
FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。
即金にて、¥1,000,000までお支払い致します。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

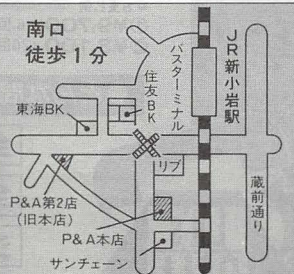
〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～60回払いまで出来ます。但し、1回のお支払い額は3,000円以上。

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

超低金利クレジット率

回数	1	3	6	10	12	15	18	24	36	48	60
利率(%)	1.5	2.0	3.0	4.5	4.5	7.5	9.0	9.5	13	17	22

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

●マイコン
●ビデオ
●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX.03-651-0141

●営業時間 AM11:00～PM9:00
日・祭日も受付けます。
(但しPM8:00迄)

なお、旧本店は、12月よりP&A第2店として、下取・買取・ソフト販売を業務に新装開店いたします。

クレジット
金利大幅
ダウン!!

話題の新製品が全国どこでも電話で買える!!

03(797)1221

COMPUTER BANK

J-DMA 安心と信頼のシステムで新時代を切り開く

X68000

"ついにベールが剥された!"68000CPU搭載。ひとつひとつのスペックに新鮮な驚きがある。未体験の機能美が創造力を刺激する。

☆注文No.A-0121

SHARP CZ-601C	¥319,800
SHARP CZ-601D	¥119,800
標準価格合計	¥439,600
現金特別価格	~~¥439,600~~

大特価にて提供中

■お支払例
①¥5,200×60回〔ボーナス〕¥13,000×10回
②¥9,200×30回〔ボーナス〕¥24,000×5回
③¥8,800×48回〔ボーナス〕無し

☆注文No.A-0122

SHARP CZ-611C	¥399,800
SHARP CZ-601D	¥119,800
標準価格合計	¥519,600
現金特別価格	~~¥519,600~~

大特価にて提供中

■お支払例
①¥6,000×60回〔ボーナス〕¥17,000×10回
②¥9,200×36回〔ボーナス〕¥26,000×6回
③¥8,900×60回〔ボーナス〕無し

☆注文No.A-1223

SHARP CZ-601C	¥319,800
SHARP CZ-601D	¥119,800
SHARP CZ-6ST1(チルトスタンド)	¥5,800
標準価格合計	¥445,400
現金特別価格	~~¥445,400~~

大特価にて提供中

■お支払例
①¥4,800×60回〔ボーナス〕¥16,000×10回
②¥8,900×30回〔ボーナス〕¥27,000×5回
③¥8,900×48回〔ボーナス〕無し

☆注文No.A-1224

SHARP CZ-611C	¥399,800
SHARP CZ-601D	¥119,800
SHARP CZ-6ST1(チルトスタンド)	¥5,800
SHARP CZ-6VT1(カラーイメージユニット)	¥69,800
標準価格合計	¥595,200
現金特別価格	~~¥595,200~~

大特価にて提供中

■お支払例
①¥7,200×60回〔ボーナス〕¥18,000×10回
②¥9,000×42回〔ボーナス〕¥29,000×7回
③¥10,200×60回〔ボーナス〕無し

当社はX68000 PRO SHOPです。

X68000オリジナルグッズ プレゼント実施中!!

X68000オリジナルグッズ(ジャンパー・マウスパッド・ポーチ)のうち、本体を御買上げのお客様にはいずれか1点、セットで御買上げのお客様には、いずれか2点をプレゼント。

X68000周辺機器・ソフトウェアリスト 大特価にて提供中

No.	型番	品名	定価	No.	型番	品名	定価	No.	型番	品名	定価
1	CZ-6VT1	カラーイメージユニット	定価 ¥69,800	9	CZ-6BN1	パラレルボード	定価 ¥29,800	17	CZ-215MS	Sampling PRO-68K	定価 ¥17,800
2	CZ-6EB1	拡張I/Oボックス	定価 ¥88,000	10	CZ-620H	20MBハードディスクユニット	定価 ¥178,000	18	CZ-220BS	DATA PRO-68K	定価 ¥58,000
3	CZ-6BE1A	1MB増設RAMボード	定価 ¥38,000	11	AN-160SP	アンプ内蔵スピーカーシステム	定価 ¥59,800	19	CZ-221HS	NEW Printshop PRO-68K	定価 ¥19,800
4	CZ-6BE2	2MB増設RAMボード	定価 ¥79,800	12	CZ-6BM1	MIDIボード	定価 ¥26,800	20	CZ-223CS	Communication PRO-68K	定価 ¥19,800
5	CZ-6ST1	チルトスタンド	定価 ¥5,800	13	CZ-211LS	Ccompiler PRO-68K	定価 ¥39,800	21	CZ-224LS	福袋 V2.0	定価 ¥9,980
6	CZ-8PC3	熱転写カラー漢字プリンタ	定価 ¥65,800	14	CZ-212BS	BUSINESS PRO-68K	定価 ¥68,000	22	CZ-226BS	CARD PRO-68K	定価 ¥68,000
7	CZ-6PV1	カラービデオプリンタ	定価 ¥198,000	15	CZ-213MS	MUSIC PRO-68K	定価 ¥18,800	23	Z's STAFF	PRO-68K(ツァイト)	定価 ¥58,000
8	CZ-8NS1	カラーイメージスキャナ	定価 ¥188,000	16	CZ-214MS	SOUND PRO-68K	定価 ¥15,800	24	Kamikaze〈神風〉	(サムシンググッド)	定価 ¥68,000

●どこよりもお得な高額下取り実施中!! セットの組合わせは自由自在、ぜひご相談下さい。

X1 turbo Z II "マルチアーティストマシン"

☆注文No.A-0125

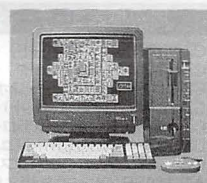
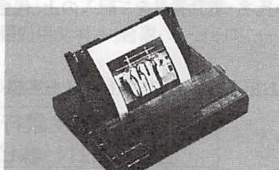
SHARP CZ-881CBK	¥179,800
SHARP CZ-880DB	¥109,800
標準価格合計	¥289,600
現金特別価格	¥220,000

■お支払例
①¥5,100×30回〔ボーナス〕¥20,000×5回
②¥9,000×18回〔ボーナス〕¥26,000×3回
③¥8,500×30回〔ボーナス〕無し

X1 twin HEシステム(PC Engine)搭載で楽しさ2倍

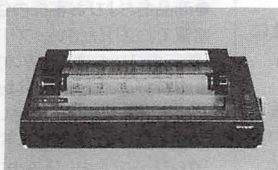
☆注文No.A-0126

SHARP CZ-830CBK	¥99,800
SHARP CZ-820DB	¥79,800
標準価格合計	¥179,600
現金特別価格	¥109,800

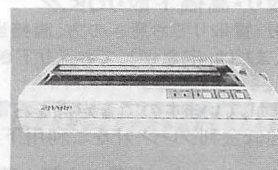
■お支払例
①¥4,800×16回〔ボーナス〕¥21,000×2回
②¥9,700×12回〔ボーナス〕無し
③¥5,200×24回〔ボーナス〕無し

●どこよりもお得な高額下取り実施中!! セットの組合わせは自由自在、ぜひご相談下さい。

☆注文No.B-0123

SHARP CZ-8PC3	¥65,800
現金特別価格	~~¥65,800~~

大特価にて提供中

■お支払例
①¥9,700×6回〔ボーナス〕無し
②¥3,100×20回〔ボーナス〕無し

☆注文No.B-0124

SHARP CZ-8PK6	¥159,000
現金特別価格	¥69,800

■お支払例
①¥7,400×10回〔ボーナス〕無し
②¥3,300×24回〔ボーナス〕無し

☆注文No.B-0125

SHARP MZ-1P17	¥79,800
CZ用ケーブル	¥7,800
標準価格合計	¥86,600
現金特別価格	¥42,800

■お支払例
①¥7,400×6回〔ボーナス〕無し
②¥3,800×12回〔ボーナス〕無し

☆注文No.B-0132

SHARP AN-8TU	¥35,800
現金特別価格	~~¥35,800~~

大特価にて提供中

●どんな問い合わせにも親切に対応いたします。

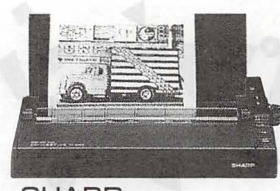
全商品保証付 中古も6ヶ月の保証期間だから安心です。	**クレジットでOK** カレッジクレジットも取扱います。
全国無料配送 お買上1万円以上、配達料はいただきません。	**日曜配達可** 留守の多い方でも安心です。
ショールーム Xシリーズ展示中。	**高額買取り** 電話1本で即、現金お支払い。
代金引換えシステム 商品到着時の代金支払いでOK。	**ボーナス一括払い** 商品は即お手元へ、お支払いはボーナス時に。

Let's CALL

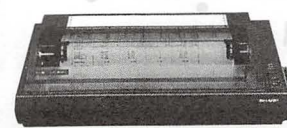
超優良中古パソコンが電話一本で買える!! 03(797)1221

SHARP
CZ-812C
(X-F／10)
¥139,800➡**¥32,000**

SHARP
CZ-820DE・B 新品
(14インチ2000字RGBTV)
¥79,800➡**¥39,800**

SHARP
CZ-880DGY 新品同様
(14インチ400/200RGBTV)
¥109,800➡**¥69,800**
(色はグレーになります。)

SHARP
CZ-8PC2 新品同様
(10インチ熱転写カラー漢字プリンタ)
¥69,800➡**¥44,800**

SHARP
CU-14A4 新品
(14インチ4050字アナログ・デジタルRGB、PC用アナログRGBケーブル付
¥89,800➡**¥49,800**

SHARP
CU-14GB/E 新品
(14インチ2000字デジタルRGB)
¥49,800➡**¥29,800**

SHARP
CZ-8PK6
(15インチ漢字プリンタ)
¥159,000➡**¥69,800** 新品
¥159,000➡**¥59,800** 新品同様

SHARP
CZ-822C
(X-1Gモデル30本体) 新品同様
¥118,000➡**¥49,800**
X-1Gモデル30TVディスプレイセット
(本体+CZ-820D) 特選極上品
¥197,800➡**¥79,800**

SHARP

本体

品名	定価	価格
CZ-801C(X-1C)	¥119,800	¥10,000
CZ-811C(X-1F model 10)	¥89,800	¥12,000
CZ-812C(X-1F model 20)	¥139,800	¥32,000
CZ-830C(X-1Twin)	¥99,800	¥52,000
CZ-880CB(X-1Turbo Z) 新品同様	¥218,000	¥74,800
MZ-1500	¥89,800	¥18,000
MZ-2521(MZ-2500 Model 30)	¥198,000	¥48,000

ディスプレイ

品名	定価	価格
12M-314C(14"カラー4050文字)	¥128,000	¥45,000
14M-131C(14"カラー2000文字)	¥69,800	¥20,000
CU-14H1(14"カラー4050文字)	¥99,800	¥45,000
CU-14H2(14"カラー4050文字)	¥99,800	¥45,000
CU-14AG2(14"カラー4050文字)	¥84,800	¥45,000
CZ-801D(14"カラー2000文字RGBTV)	¥99,800	¥30,000
CZ-880D(14"カラー2000文字RGBTV)	¥109,800	¥64,800
CU-14BD(14"カラー4050文字)	¥64,800	¥40,000

ディスクドライブ・プリンタ・他

品名	定価	価格
MZ-1F07(5"2D、2ドライブ)	¥158,000	¥38,000
CZ-81P(ミニサイズプリンタ)	¥34,800	¥10,000
CZ-8PP2(カラープロッタプリンタ) 新品	¥54,800	¥15,000
CZ-8PD2(10"ドットプリンタ)	¥79,800	¥28,000
CZ-8PD3(10"ドットプリンタ)	¥59,800	¥28,000
CZ-8PC2(10"24ドット漢字熱転写プリンタ) 新品	¥69,800	¥44,800
CZ-8PK6(15"24ドット漢字プリンタ) 新品同様	¥159,000	¥59,800
MZ-80BP5(80桁プリンタ)	¥142,000	¥18,000
MZ-1P06(80桁漢字プリンタ)	¥234,000	¥45,000
MZ-1P09(MZ-1500用カラープロッタプリンタ) 新品	¥47,600	¥25,800
MZ-1P17(80桁24ドットカラー漢字熱転写プリンタ・CZ用ケーブル付) 新品	¥76,600	¥42,800
MZ-1P17(80桁24ドットカラー漢字熱転写プリンタ・MZ用ケーブル付) 新品	¥76,600	¥46,800
CZ-8SS2(システムスタンド)	¥5,500	¥4,000
CZ-8BS1(FM音源ボード) 新品	¥23,800	¥20,000

＊SHARP X-1シリーズ特選極上品コーナー＊

品名	定価	価格
CZ-820CE(X-1G/10) 新品同様	¥69,800	¥16,800
CZ-822CB(X-1G/30) 新品同様	¥118,000	¥49,800
CZ-822C+CZ-822D(X-1G/30セット) 特選極上品	¥197,800	¥79,800

＊SHARP ディスプレイ特選極上品コーナー＊

品名	定価	価格
CU-14G(14"カラー2000文字) 新品	¥49,800	¥29,800
CU-14A4(14"カラー4050文字) 新品	¥89,800	¥49,800
CZ-820D(14"カラー2000文字RGBTV) 新品同様	¥79,800	¥39,800
CZ-880DB(15"カラー4050文字RGBTV) 新品同様	¥109,800	¥85,000
CZ-880DGY(15"カラー4050文字RGBTV) 新品同様	¥109,800	¥69,800
CZ-600D(15"カラー4050文字RGBTV) 新品同様	¥129,800	¥88,000

6つの安心のアフターサービス

1 C.B.クラブ

■あなたも今すぐ会員に!!

当社で商品をお買い上げの方全員に、C.B.クラブカードを無料でお送り致します。このカードをお持ちの方なら次の買い換え時や、付属品の購入時に会員特別価格でご購入になれます。

2 C.B.サポートホットライン

☎03(797)1234

■トラブルへの対応!!

当社でコンピュータをお買い上げいただいたお客様に万一、トラブルが発生した場合、このホットラインで親切に対応いたします。

3 C.B.レスキューシステム

■迅速なサポート体制!!

お客様のお手元でトラブルが発生した場合、当社より引取りにお伺い致します。万一、お買いになった機械が故障しても安心です。

4 C.B.クイック・チェンジシステム

■新品交換体制も万全!!

お買い上げになったパソコンが、万一初期不良でも安心です。商品到着後7日以内にご連絡いただければ、新品と交換致します。

QUICK CHANGE

5 PC-9801VX21アフターサポート

■VX21愛好家にお得です!!

NEC PC-9801VX21をお買い上げいただいたお客様に保証期間中、万一故障があった場合無料で代品を貸出します。

6 C.B.Q&Aホットライン

☎03(797)1233

■素朴な疑問何でもどうぞ!!

ハードウェア、ソフトウェアに関するご質問なら内容を問わずどなたからでも親切に、ご相談をお受け致しております。

- 電話一本で高額下取り、即商品はお手元へ!
- あなたの不要になったパソコンを電話一本で査定し買取ります。
- 掲載の商品以外も取り扱っておりますのでお気軽にお電話下さい。

▼本社注文デスク
03(797)1221

コンピュータバンク

株式会社パシフィックコンピュータバンク　〒150 東京都渋谷区渋谷1-6-8 井上ビル　営業時間／AM9:30〜PM9:30 年中無休

パソコン・AV専門 O.A.ランド

大特価セール

セール期間 '88 12・16→'89 1・15

安心と信頼のOAランド・優良パソコン販売店
アフターサービス万全のサポート体制

ランド特選!! SHARP X68000 ACE-HD / ACE セット

Ⓐ ACE-HD セット
- CZ-611C……………………定価￥399,800
- CZ-611D……………………定価￥145,000
- CZ-6ST1……………………定価￥ 5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

合計価格￥550,600

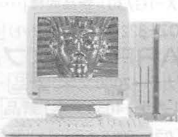
現金大特価!! 安いぞ

★ランドクレジット例
①12回 ￥34,900 ②24回 ￥19,100 ③36回 ￥12,900

Ⓑセット ACE-HD セット
- CZ-611C……………………定価￥399,800
- CZ-601D……………………定価￥119,800
- CZ-6ST1……………………定価￥ 5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

合計価格￥525,400

現金大特価!! お買得

★ランドクレジット
①12回 ￥34,900 ②24回 ￥18,400 ③36回 ￥12,900

Ⓒ ACE-HD セット ディスプレイCu-21CD
- CZ-611C……………………定価￥399,800
- CU-21CD……………………定価￥130,800
  (21インチ3モードスキャン)
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

合格価格￥539,600

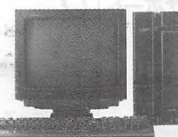
現金大特価!! 注目

★ランドクレジット
①12回 ￥35,200 ②24回 ￥18,600 ③36回 ￥13,000

Ⓓ ACE セット
- CZ-601C……………………定価￥319,800
- CZ-601D……………………定価￥119,800
- CZ-6ST1……………………定価￥ 5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

合計価格￥445,400

現金大特価!! 推選

★ランドクレジット
①12回 ￥28,600 ②24回 ￥151,00 ③36回 ￥10,600

Ⓓ ACE セット
- CZ-601C……………………定価￥319,800
- CZ-611D……………………定価￥145,000
- CZ-6ST1……………………定価￥ 5,800
- MD-2HD 20枚サービス
- 市販ゲームソフト2本サービス

合計価格￥470,600

現金大特価!! お値打品

★ランドクレジット
①12回 ￥30,400 ②24回 ￥16,000 ③36回 ￥11,200

X-1ターボZⅢセットNEW

Ⓐセット
- CZ-881CBK…定価￥169,800
- CZ-880DBK…定価￥ 99,800
- CZ-6ST1-B…定価￥ 5,800
  (チルトスタンド)
- MD-2HD 20枚サービス

合計定価￥275,400

現金価格
特価中TEL下さい

★ランドクレジット
①12回 ￥21,000
②24回 ￥11,000
③36回 ￥ 7,300

CRTクリーナー
キーボードカバー
プレゼント!!

Ⓑセット
- CZ-888CBK…定価￥169,800
- CZ-830DBK…定価￥ 98,000
- CZ-6ST-1B…定価￥ 5,800
  (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥273,600

合計価格
特価中TEL下さい

★ランドクレジット
①12回 ￥15,700
②24回 ￥ 8,300
③36回 ￥ 5,800

X-1ターボⅡセット

Ⓐセット
- CZ-881CBK…定価￥179,800
- CZ-880DBK…定価￥109,800
- CZ-6ST1-B…定価￥ 5, 800
  (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥295,400

現金価格
￥198,000

★ランドクレジット
①12回 ￥15,100
②24回 ￥ 7,900
③36回 ￥ 5,600

CRTクリーナー
キーボードカバー
プレゼント!!

Ⓑセット
- CZ-881CBK…定価￥179,800
- CZ-830BK…定価￥ 98,000
- CZ-6ST1-B…定価￥ 5,800
  (チルトスタンド)
- MD-2HD 20枚サービス

合計価格￥283,600

現金価格
￥178,000

★ランドクレジット
①12回 ￥14,200
②24回 ￥ 7,500
③36回 ￥ 5,200

X-1Gセット

X-1Gセット お買得
- CZ-822CB…定価￥118,000
- CZ-820DB…定価￥ 79,800
- MD-2D 20枚サービス

合計価格￥197,800

①X-1Gセット＋
パソコンラック
(BW-OU 定価￥19,500)

現金特価 **￥81,000**

②X-1Gセット＋
市販ゲームソフト2本

現金特価 **￥79,800**

70,000人もの人々が体感した安心感。——信頼のIPLワイドサポート

●業界初、IPLでこそ成し得た3倍保証。

メーカー保証12ヶ月の商品なら36ヶ月の保証と長期間の保証を実施。末長く安心してご利用いただけるよう、IPLが成し得たワイドなサポート体制。

●IPLだからこそ初期不良への保証も万全。交換期間も1ヶ月ともっとも長期間です。IPLだからこそ安心が長続きします。

こんなにかかる修理費用

プリンタヘッド交換¥29,500以上/98シリーズメインボード交換¥21,600以上/ドライブ交換¥13,200以上

比べてほしいから、ご紹介します。——さらにお買得IPLクレジット

●ステップアップクレジットがおトク。

まず月々1,000円からスタートして2年後から3,000円アップ。ボーナスも1年後1万円。3年後3万円。また夏のボーナスを貯金して冬のボーナスからのお支払いも大丈夫。夏・冬のボーナスどちらか一つをセレクト。ボーナス年一回だけもOK。

システムはすぐお手元へ。冬のボーナス一括、冬夏ボーナス2括払いもOK。

●追加購入もクレジットだから便利。

追加購入も買い換えもご利用中のIPLクレジットを月々僅か1,000円ずつの調整でOK。

●IPLは8月21日ついに70,001人めのお客様を迎えました。

ORDER TELEPHONE

電話受付:AM10:00～PM8:00 水曜定休日

本社 0467-24-7511
大阪 06-311-2736

銀座 03-541-3058　青山 03-470-0061　札幌 011-621-1444
仙台 022-266-0531　広島 082-293-7881　福岡 092-481-2644

商品管理部(納期、配達日のお問合せ、ご指定日のご連絡) 0467-24-1154
メンテナンス部(ハード上のご相談、お問合せ、初期不良の対応) 0467-24-0453
FAX(ご注文、お見積り、カタログ編集などスピーディに) 0467-24-0561
タイムリーボックス(ホットな新製品ニュースをお知らせします。) 0467-24-0941

ご注文お問合せ ——0467-24-1154
下取りホットライン ——0467-24-2040

CHANCE

1. 期間中、システムでお買い上げの方、先着200名様に、**電話帳電卓**をプレゼント。(電話番号・スケジュールを記憶、10桁電卓機能付)
2. 期間中、デスク(SA-600)をお買上げの方全員に、A-300(**原稿用スタンド**¥8,000)をプレゼント。
3. 期間中、シャープ製品をシステムでお買上げの方にCZ-8NJ1(**ジョイカード**)をプレゼント。
4. 期間中、エプソンAP800お買い上げの方全員に**リボンパック**(金2本銀1本)をプレゼント
5. 期間中、エプソンAP-800(シャープ用)をお買上げの方全員に**ケーブル**(¥8,800)得々パックをプレゼント
期間中、セットでお買上げのご希望の方に**モデム**(SR-30 ¥19,800)をプレゼント

SHARP X68000

アクセス No.X0181

価格¥1,184,800 ➡ IPL超特価

品名	価格
CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(,39ミリ、アナログ3モードオートスキャン)	¥119,800
CZ-6ST1(角度自由自在、調節OK!)	¥5,800
CZ-6VT1(カラーイメージユニット、テロッパー機能付き)	¥69,800
CZ-6PV1(カラービデオプリンタ)	¥198,000
CZ-8NS1(フルカラーA4ズーム機能色ずれの少ない線順次方式ソフト付き)	¥188,000
CZ-6BN1(68000用スキャナ用パラレルボード)	¥29,800
AN-160SP(アンプ内蔵スピーカーシステム2本組)	¥59,800
C-TRACE(CGアニメーション用ソフト)	¥68,000
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥65,800
CZ-232AS(熱血高校ドッジボール部)	¥7,800
CZ-218AS(沙羅曼蛇)	¥8,800
ラストハルマゲドン	¥9,800
こはく色の遺言	¥9,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
電話帳電卓*贈呈(電話番号50人分,スケジュルメモOK電卓機能付)	プレゼント中
CZ-8NJ1(ジョイカード)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
SR-30用ケーブル(¥5,000)	プレゼント中
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き。解り易い解説)	¥0
初期不良期間(ワイドに1ヶ月間の交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

標準価格¥1,184,800

月々	ボーナス
¥7,700×72回	5.0万×12回
¥10,000×72回	3.58万×12回
¥10,200×60回	5.0万×10回
¥13,900×48回	5.0万×8回
¥20,300×36回	5.0万×6回

夏のボーナス一括
システムはお手元へ

アクセス No.X0182

価格¥683,500 ➡ IPL超特価

品名	価格
CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(,39ミリ、アナログモードオートスキャン)	¥119,800
CZ-6ST1(角度自由自在、調節OK!)	¥5,800
CZ-213MS(MUSIC PRO 68K)	¥18,800
CZ-214MS(SOUND PRO 68K)	¥15,800
CZ-226BS(ワープロ機能を備えたカード型リレーショナルデータベース)	¥29,800
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥65,800
CZ-6SD1(X68専用キャスター、スライドテーブル付キーボード収納OK)	¥44,800
SNC-081(布張肘掛け付き回転イス)	¥20,500
信長の野望/全国版(光栄5"2HD)	¥9,800
ドラゴンスピリット(マイコンソフト)	¥8,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
電話帳電卓*贈呈(電話番号50人分,スケジュルメモOK電卓機能付)	プレゼント中
CZ-8NJ1(ジョイカード)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
SR30用シャープケーブル(¥5,000)	プレゼント中
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き。解り易い解説)	¥0
初期不良期間(ワイドに1ヶ月間の交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

標準価格¥683,500

月々	ボーナス
¥3,000×72回	3.62万×12回
¥5,000×72回	2.42万×12回
¥6,600×54回	3.0万×9回
¥8,000×42回	3.8万×7回
¥10,000×36回	3.73万×6回

(¥) 月々わずか1000円

超低金利……　組合せ自由　全国無料配送

※今回掲載の製品は、12月18日より1月18日までの期間に限らせていただきます。

アクセス No.X0180

価格¥511,400 ➡ IPL超特価

品名	価格
CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(.39ミリ、アナログ3モードオートスキャン)	¥119,800
CZ-6ST1(角度自由自在、調節OK!)	¥ 5,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
源平討魔伝	¥ 7,800
信長の野望/全国	¥ 9,800
スペースハリアー	¥ 6,800
CZ-218AS(沙羅曼蛇)	¥ 8,800
ドラゴンスピリット	¥ 8,800
CZ-8NJ1(ジョイカード)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
SR30用シャープケーブル(¥5,000)	プレゼント中
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き、解り易い解説)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥3,000 ×72回 ボーナス 2.08万×12回 標準価格¥511,400

月額		ボーナス
¥ 6,500×72回	ボーナス	なし
¥ 3,300×54回	ボーナス	3.0万×9回
¥ 6,700×36回	ボーナス	3.0万×6回
¥10,000×24回	ボーナス	4.18万×4回

◆初期不良交換期間1ヶ月◆

SHARP X68000

アクセス No.X0187

価格¥105,600 ➡ IPL超特価

品名	価格
CZ-211LS(C compilerソフト開発を効率良く(サポート))	¥ 39,800
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥ 65,800
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥1,800 ×60回 ボーナス なし 標準価格¥105,600

月額		ボーナス
¥ 2,800×36回	ボーナス	なし

◆日曜・祭日指定配達◆

SHARP X68000 ACE HD

アクセス No.X0183

価格¥710,400 ➡ IPL超特価

品名	価格
CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
CZ-6ST1(角度自由自在、調節OK!)	¥ 5,800
CZ-213MS(MUSIC PRO 68K)	¥ 18,800
CZ-214MS(SOUND PRO 68K)	¥ 15,800
CZ-215MS(AD PCM機能をサポートしたサンプリングエディタ)	¥ 17,800
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥ 65,800
CZ-218AS(沙羅曼蛇)	¥ 8,800
ドラゴンスピリット(マイコンソフト)	¥ 8,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
電話帳電卓(電話番号50人分、スケジュールメモOK電卓機能付)	プレゼント中
CZ-8NJ1(ジョイカード)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
#8226(SR30用シャープケーブル)	プレゼント中
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き、解り易い解説)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥3,000 ×72回 ボーナス 3.78万×12回 標準価格¥710,400

月額		ボーナス
¥ 5,000×72回	ボーナス	2.57万×12回
¥ 6,900×54回	ボーナス	3.0万×9回
¥ 8,000×42回	ボーナス	4.0万×7回
¥ 8,400×36回	ボーナス	5.0万×6回

アクセス No.X0184

価格¥800,200 ➡ IPL超特価

品名	価格
CZ-611C(20MHDD搭載、65536同時発色、FM8音源内蔵)	¥399,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
CZ-211LS(C compilerソフト開発を効率良く(サポート))	¥ 39,800
Z'sSTAFF PRO 68K(グラフィックツール)	¥ 58,000
CZ-226BS(ワープロ機能を備えたカード型リレーショナルデータベース)	¥ 29,800
CZ-6BE1A(1MB増設RAMボード)	¥ 38,000
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥ 65,800
3Mブランクディスケット(5"2HD*10枚)	プレゼント中
電話帳電卓(電話番号50人分、スケジュールメモOK電卓機能付)	プレゼント中
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
SR30用シャープケーブル(¥5,000)	プレゼント中
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き、解り易い解説)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥2,200 ×72回 ボーナス 5.0万×12回 標準価格¥800,200

月額		ボーナス
¥ 5,000×72回	ボーナス	3.28万×12回
¥ 7,200×60回	ボーナス	3.0万×10回
¥ 9,600×48回	ボーナス	3.0万×8回
¥10,500×36回	ボーナス	5.0万×6回

70,000人もの人々が体感した安心感。
———信頼のIPLワイドサポート

アクセス No.X0185

価格¥667,600 ➡ IPL超特価

品名	価格
CZ-611C(20MHDD搭載、65536同時発色、FM8音源内蔵)	¥399,800
CZ-603D(高解像度0.31ドットピッチ、オーバースキャン、チルト付)	¥ 84,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
CZ-8PK6(24ピン136桁漢字プリンタ)	¥159,000
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント中
SR30用シャープケーブル(¥5,000)	プレゼント中
電話帳電卓(電話番号50人分、スケジュールメモOK電卓機能付)	プレゼント中
添削付通信講座(信頼の"サポート"テスト問題付 ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥3,000 ×72回 ボーナス 2.72万×12回 標準価格¥667,600

月額		ボーナス
¥ 4,200×72回	ボーナス	2.0万×12回
¥ 4,700×54回	ボーナス	3.0万×9回
¥ 6,900×42回	ボーナス	3.0万×7回
¥ 7,200×36回	ボーナス	3.8万×6回

組み合せ自由

アクセス No.X0189

価格¥790,000 ➡ IPL超特価

品名	価格
CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-611D(.31ミリ、アナログ3モードオートスキャン)	¥145,000
CZ-6ST1(角度自由自在、調節OK!)	¥ 5,800
CZ-6VT1(カラーイメージユニット、テロッパー機能付き)	¥ 69,800
CZ-6BC1(I/Oスロットに装着パソコンからFAX送受信A4B4GIII)	¥ 79,800
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥ 65,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
電話帳電卓(電話番号50人分、スケジュールメモOK電卓機能付)	プレゼント
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19800)	プレゼント
SR30用(ケーブル¥5000)	プレゼント
X68通信講座(信頼の"サポート"テスト問題付ひとりひとりをしっかりフォロー)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム!)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0

¥1,900 ×72回 ボーナス 5.0万×12回 標準価格¥790,000

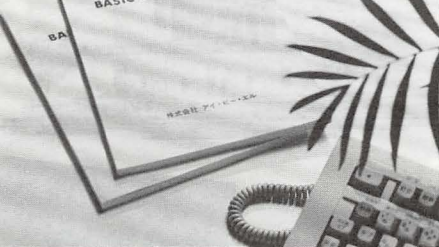
月額		ボーナス
¥ 5,000×72回	ボーナス	3.15万×12回
¥ 8,100×54回	ボーナス	3.0万×9回
¥10,000×42回	ボーナス	3.69万×7回
¥10,100×36回	ボーナス	5.0万×6回

COMPUTER CREATION

IPL

株式会社・アイピーエル
〒248 鎌倉市雪ノ下4-1-12
雪ノ下ビル
鎌倉市雪ノ下3-4-23:商品管理部
AM10:00▶PM8:00
水曜日定休

安心の3倍保証

ご好評の通信教育。添削付テスト問題も加わり、より深く、よりきめ細かくフォローします。

IPL
実績から実戦・
X68初の通信教育制度

アクセス No.X0188

価格¥662,200 ➡ IPL超特価

品名	価格
CZ-611C(20MHDD搭載、65536色発色、FM8音源内蔵)	¥399,800
CZ-603D(超高解像度0.31ドット14"チルト台付)	¥ 84,800
Z'sSTAFF PRO 68K(グラフィックツール)	¥ 58,000
CZ-226BS(ワープロ機能を備えたカード型リレーショナルデータベース)	¥ 29,800
CZ-8PC3(10"カラー熱転写、ハガキ可、漢字53字/秒)	¥ 65,800
3Mブランクディスケット(5"2HD*10枚)	¥ 24,000
電話帳電卓*贈呈(電話番号50人分、スケジュールメモOK 電卓機能付)	¥ 0
X68通信講座(業界初!信頼のオリジナル"サポート"添削付き、解り易い解説)	¥ 0
初期不良期間(ワイドに1ケ月間の交換システム)	¥ 0
安心の3倍保証(IPL保証書付き)	¥ 0
パーソナルモデムSR-30(パソコンの新しい可能性をプレゼント¥19,800)	プレゼント

¥3,000 ×72回 ボーナス 3.64万×12回 標準価格¥662,200

月額		ボーナス
¥ 5,000×72回	ボーナス	2.45万×12回
¥ 6,700×54回	ボーナス	3.0万×9回
¥ 9,300×42回	ボーナス	3.0万×7回
¥10,000×36回	ボーナス	3.8万×6回

アクセス No.X0186

価格¥45,000 ➡ IPL超特価

品名	価格
源平討魔伝	¥ 7,800
CZ-218AS(沙羅曼蛇)	¥ 8,800
ドラゴンスピリット	¥ 8,800
ラストハルマゲドン	¥ 9,800
信長の野望/全国	¥ 9,800

¥1,400 ×36回 ボーナス なし 標準価格¥45,000

月額		ボーナス
¥ 2,000×24回	ボーナス	なし

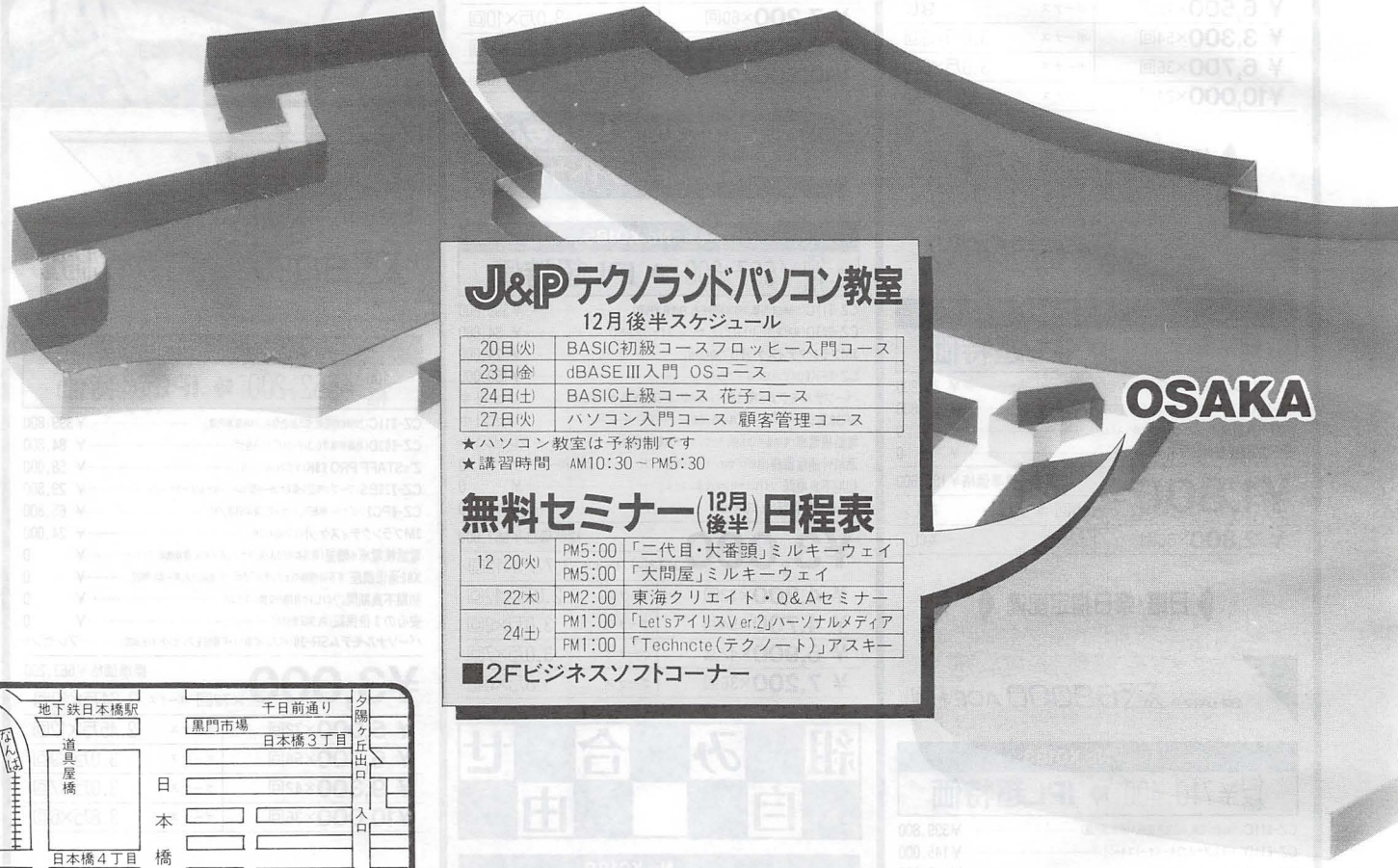
翌月1ヶ月から自由に設定
'89・8月からお支払OK

日曜・祭日・指定日配送OK! 輸送上のトラブルにも対応

※IPL超特価システムはさらにおトクなシステム。お電話でお問合せください。

Dec.18〜Jan.18

大阪の人も、パソコンなら、

OSAKA

J&P テクノランドパソコン教室

12月後半スケジュール

20日(火)	BASIC初級コースフロッピー入門コース
23日(金)	dBASEⅢ入門 OSコース
24日(土)	BASIC上級コース 花子コース
27日(火)	パソコン入門コース 顧客管理コース

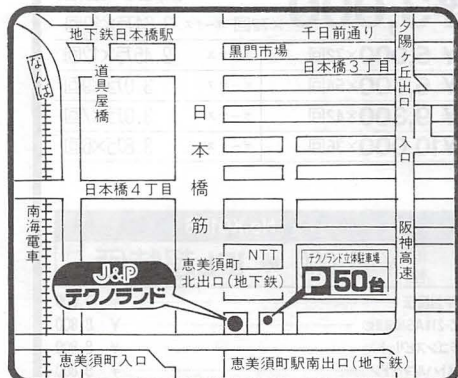
★パソコン教室は予約制です
★講習時間 AM10:30～PM5:30

無料セミナー(12月後半)日程表

12 20(火)	PM5:00	「二代目・大番頭」ミルキーウェイ
	PM5:00	「大問屋」ミルキーウェイ
22(木)	PM2:00	東海クリエイト・Q&Aセミナー
24(土)	PM1:00	「Let'sアイリスVer.2」パーソナルメディア
	PM1:00	「Technote(テクノート)」アスキー

■2Fビジネスソフトコーナー

Joshin Computer Store
J&P テクノランド

大阪市浪速区日本橋5丁目6番7号(〒556)
☎06(634)-1211

最先端情報の提案と万全

ハードウェア

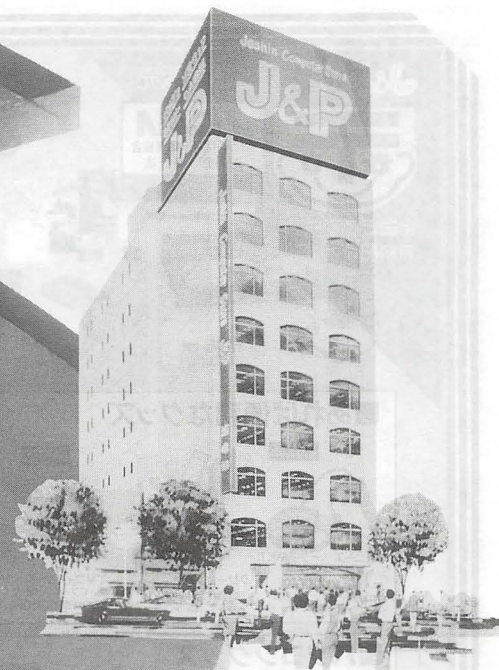
MSXからIBMまで人気のハードウェアを一堂に集結。プリンタ、ハードディスクなどの周辺機器もズラリ勢揃い。さまざまなニーズに的確にお応え出来る品揃えです。

ソフトウェア

人気のホビーソフトをはじめ、日本語ワープロ、簡易言語、販売管理、顧客管理、CADシステム、データベース、各種OSなどあらゆる用途のソフトウェアを取り揃えました。

システムサポート

職場のOA化を図るためのコンサルティングはもちろん、お客様のビジネスにピッタリのソフトウェアを設計制作いたします。

東京の人も、J&Pです。

TOKYO

1F イベントごあんない

タ方フェア ゲームソフト実演デモ PM3:30～6:30

日付	内容
12 19(月)	「ディアブロ」ブローダーボンド
21(水)	「麻雀武蔵」「キングオブマジック」コスモスコンピュータ
22(木)	「プロダクションマネー」コムパック
23(金)	「プロダクションマネー」コムパック

ダイナウェアミュージックソフト(98用)店頭実演デモ PM1:00～6:00

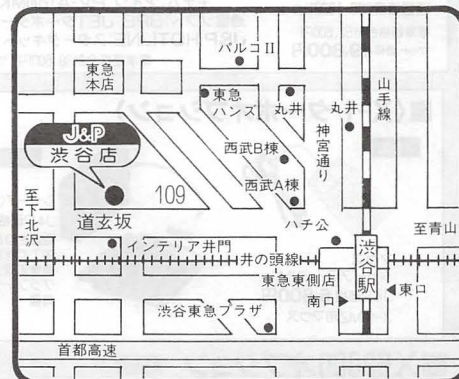
「プレリュード」「バラード」「ミュージ君」

ご購入特典 ●「バラード」…………特制クリスマスソング集
●「ミュージ君」……ミュージ君BOOK & オリジナルデータ集

ご来場記念 ●プレリュードオートデモプレゼント(先着30名様)

3F ビジネスソフト店頭実演デモ日程 PM1:00～5:00

日付	内容
12 19(月)	「TOP財務会計・給与計算」オービックビジネスコンサルタント
	「弥生」日本マイコン販売
20(火)	「二代目大番頭・大福帳」ミルキーウェイ
	「Let's アイリス」パーソナルメディア
21(水)	「隼(HAYABUSA」「1・2・3(ARd)」ダットジャパン
	「dBASEIII PLUS」日本アシュトンテイト
	「PI EXE」dBソフト
22(木)	「経理部長」テクニカルソフト
	「Gennereic CADD Let's13」ソフトウェアジャパン
23(金)	「VP-Planner」東海クリエイト
	「DYNAPERS 3」「UPシリーズ八方美人」ダイナウェア
	「毛筆ワープロJr」富士ソフトウェア

のサポート体制を誇るJ&P

Joshin Computer Store
J&P 渋谷店
東京都渋谷区道玄坂2丁目28番4号(〒150)
☎(03)496-4141

クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(年中無休)
AM10:00～PM7:00(日曜・祭日はPM6:00まで)

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

(★X68000をお買上げのお客様にもれなく、▶X68000オリジナルテレホンカードプレゼント!!)

基本セット X68000 ACE

品目	価格
●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-6ST1(チルトスタンド)	¥ 5,800
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
●ソフト/アルカノイド	¥サービス
■定価合計	¥455,400▶クリエイト特価

均等払い	¥10,280×24回	¥ 6,350×36回	¥ 4,830×48回
ボーナス	¥30,000× 4回	¥25,000× 6回	¥20,000× 8回

Vセット X68000 ACE

品目	価格
●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-8PC2(熱転写カラー漢字プリンタ)	¥ 65,800
●CZ-6TV1(カラーイメージユニット)	¥ 69,800
●CZ-6ST1(チルトスタンド)	¥サービス
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
●ソフト/アルカノイド	¥サービス
■定価合計	¥585,200▶クリエイト特価

均等払い	¥13,180×24回	¥ 8,650×36回	¥ 6,440×48回
ボーナス	¥40,000× 4回	¥30,000× 6回	¥25,000× 8回

大サービス・ゲーマーズセット X68000 ACE

品目	価格
●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●ドラゴンスピリッツ	¥ 8,800
●沙羅曼蛇	¥ 8,800
●XE-1 PRO(ジョイスティック)	¥ 9,800
●ドッジボール	¥サービス
●アルカノイド	¥サービス
●CZ-6ST1(チルトスタンド)	¥サービス
■定価合計	¥467,000▶クリエイト特価

均等払い	¥ 9,900×24回	¥ 6,390×36回	¥ 5,510×48回
ボーナス	¥40,000× 4回	¥30,000× 6回	¥20,000× 8回

基本セット X68000 ACE HD

品目	価格
●CZ-611C(本体+キーボード)	¥399,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-6ST1(チルトスタンド)	¥ 5,800
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
●ソフト/アルカノイド	¥サービス
■定価合計	¥535,400▶クリエイト特価

均等払い	¥13,700×24回	¥ 8,700×36回	¥ 6,660×48回
ボーナス	¥30,000× 4回	¥25,000× 6回	¥20,000× 8回

ミュージックワークセット X68000 ACE

品目	価格
●CZ-601C(本体+キーボード)	¥319,800
●CZ-601D(カラー専用ディスプレイ)	¥119,800
●CZ-8PC3(熱転写カラー漢字プリンタ)	¥ 65,800
●SOUND PRO-68K(音色作成ツール)	¥ 15,800
●CZ-6ST1(チルトスタンド)	¥ 5,800
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
●MUSIC PRO-68(楽譜入力ツール)	¥サービス
■定価合計	¥537,000▶クリエイト特価

均等払い	¥11,580×24回	¥ 7,550×36回	¥ 5,580×48回
ボーナス	¥40,000× 4回	¥30,000× 6回	¥25,000× 8回

グラフィックワークセット X68000 ACE HD

品目	価格
●CZ-611C(本体+キーボード)	¥399,800
●CZ-611D(0.31ピッチ・カラーディスプレイ)	¥145,000
●CZ-6PV1(カラービデオプリンタ)	¥198,000
●Z's STAFF PRO-68K	¥ 58,000
●レイトレーシングソフト	¥ 68,000
●CZ-6ST1(チルトスタンド)	¥サービス
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
■定価合計	¥878,800▶クリエイト特価

均等払い	¥23,050×24回	¥14,930×36回	¥11,770×48回
ボーナス	¥50,000× 4回	¥40,000× 6回	¥30,000× 8回

新品超お買得品セット

品目	価格
●CZ-820CE	¥ 69,800
●CZ-820DE	¥ 79,800
●CZ-503F(5インチシングルドライブ)	¥ 49,800
■定価合計	¥199,400

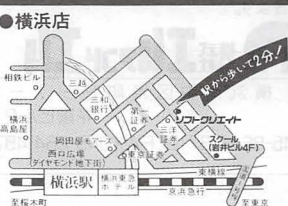
大特価 ¥78,800

基本セット X1 turbo Z III

品目	価格
●CZ-888CBK(本体+キーボード)	¥169,800
●CZ-860DBK(カラーディスプレイ)	¥ 99,800
●CZ-6ST1(チルトスタンド)	¥ 5,800
●ブランクディスケット(5″2HD・10枚)	¥ 10,000
■定価合計	¥285,400▶クリエイト特価

均等払い	¥ 6,020×24回	¥ 3,940×36回	¥ 3,330×48回
ボーナス	¥20,000× 4回	¥15,000× 6回	¥10,000× 8回

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。電話にてお問合せください。

年末年始SHARP 特別感謝セール!!
12月15日(木)～1月15日(日)
- ▶新品特価セール!
- ▶ゲームソフト20%OFF!
- ▶周辺機器、サプライ用品大特価!

X68000シリーズ用 周辺機器お買い得セール

型番	品名	定価	特価	型番	品名	定価	特価
CZ-6VT1	カラーイメージユニット	¥ 69,800	クリエイト特価	CZ-6EB1	拡張I/Oボックス(4スロット)	¥ 88,000	クリエイト特価
CZ-8NS1	カラーイメージスキャナ	¥188,000		CZ-6BC1	FAXボード	¥ 79,800	
CZ-6BE1A	1MB増設RAMボード	¥ 38,000		CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	
CZ-6BE2	2MB増設RAMボード	¥ 79,800		CZ-8BS1	ステレオFM音源ボード	¥ 23,800	
CZ-6BE4	4MB増設RAMボード	¥138,000		CZ-603D	ドットピッチ0.31㎜ 14型高解像度	¥ 84,800	
CZ-6BU1	ユニバーサルI/Oボード	¥ 39,800		CU-14CD	ドットピッチ0.31㎜ 14型高解像度	¥ 84,800	
CZ-6BG1	GP-IBボード	¥ 59,800		CU-14ED	ドットピッチ0.39㎜ 14型高解像度	¥ 79,800	
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800		AN-8TU	パソコンチューナ	¥ 35,800	

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:協和銀行 渋谷支店(普)No.239313

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

ACCESS

X1 エミュレータ

定価￥9,800

●X68000でX1を体験したい君に！

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するソフトエミュレータです。X68000上に実現した仮想X1マシンをお楽しみ頂けます。

●X1ソフトをX68000で遊びたい君に！

X1ソフトをX68000上にファイル転送できますので、これまでにX1で作った多くのプログラムをX68000で体験できます。

●やっぱりX1がかわいい君に！

X68000を使いながらもX1を使っている気持ちになれます。

実行可能アプリケーションソフト

●HuBASIC ●X1 CP／M ●X1 LOGO
【X1 CP／M用 ランゲージシリーズ】●APL ●LISP ●COBOL ●C ●FORTH ●FORTRAN ●PASCAL
●etc（X1シリーズ用とされているものに限ります。）

＊プロテクトの施してあるソフトは実行できません。
＊一部サポートしていない機能があります。
＊タイミング等ハードウェアに依存するようなものは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
＊実行速度はX1と比較して約1／3〜1／5になります。

X1 5″ 2D ⟺ X68000 Human68k

ファイル転送ユーティリティ

ディスク転送 X1ディスク↔X68000 Human68k（5″ 2Dディスクイメージファイル）
X1エミュレータはHuman68kディスク上のX15″ 2Dディスクイメージファイルを仮想ドライブとして使用！

ファイル転送 X1 BASIC↔X68000 Human68k：X1 CP／M↔X68000 Human68k
X1で作ったプログラム＆データがX68000で使える！

メニューで実行 このユーティリティはメニューにそって実行するので操作は簡単！

＊ファイルを転送するために専用ケーブルが付属します。X1とX68000をつないでご使用ください。

★X1エミュレータの購入方法　このほど、このX1エミュレータは、直接弊社よりみなさまにお届けできるようになりました。詳しくはお問合せください。

MS-DOS エミュレータ

CONCERTO-X68K

定価￥99,800

CONCERTO-X68K（コンチェルト）はX68000上でお使い頂くMS-DOSエミュレータです。専用ハードウェア：DOS Engineとエミュレーションソフトで構成され、特定機種専用のものを除くMS-DOS V2.11のソフトがX68000上でお使い頂けます。DOS EngineはNEC V30 CPUを使用しており、MS-DOSソフトの高速実行を実現しております。1台のマシンで全く異なるハードをコントロール。X68000自身の持つ高速ディスクアクセス等の優れた性能をいかし、使い慣れたMS-DOSソフトをそのままご利用頂けます。これによりX68000の世界がさらに広がります。

専用ハード：DOS Engine

●8MHzのV30を使用（メモリノーウェイト）
●ボード上にMS-DOSの実行用メモリ512KByte搭載
●数値演算プロセッサ8087-1実装可能（オプション）

＊ボードは本体より12cm程度大きくなります。その部分にはカバーがつきます。

MS-DOS用実行可能アプリケーションソフト

●MS-C（Ver3.00，4.00）
●MS-FORTRAN（Ver3.13，4.01）
●MS-PASCAL（Ver3.13）
●MS-LINK（Ver2.01，2.20，2.44）
●Lattice C（Ver2.12，3.10）
●Optimizing-C（Ver2.20F）
●TURBO PASCAL（Ver2.00B，3.01A）
●Plink 86（Ver1.46） etc……

（実行可能ソフトの一例です。）

代理店募集

アクセスではこれらの製品の発売にあたり代理店を募集しております。詳しくはお問い合せください。

有限会社アクセス　〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03（233）0200㈹ FAX.03（291）7019

資料請求券 oh!X 1月号

通信生活！はじめます

きっと、出会える。キット で、会える。

"スタータキット"で新しいくらしに出会えます。

新しい年に不思議な出会い。あなたのパソコンやワープロを電話線とドッキングさせれば、もうパソコン通信のはじまりです。画面にあらわれるのは、各種のニュースや、愉快な仲間のメッセージ。いままで体験したこともない、新しいくらしに出会えます。パソコン通信ならJ&P HOTLINE。スタータキットがご案内します。

- ***DATABASE***/パソコン・ワープロを図書館として使う楽しさ。
- ***電子メール***/パソコン・ワープロが留守番電話になる便利さです。
- ***BBS***/見知らぬ仲間と知り合うチャンスが生まれます。
- ***SIG***/スペシャリストが大集合。専門知識が豊富になります。
- ***SHOPPING***/会員割引がお得です。

1か月の試用
アクセス可能！

パソコン通信知ってみたい！

スタータキットでわかります。

パソコン通信の仕方を楽しく解説したマニュアル「パソコン通信を楽しむ本」をはじめ、初心者の方向けに大切な情報をぎっしりと詰め込んだ「スタータキット」。ID番号やパスワードも同封していますから、買ったその日からアクセス開始！1ヵ月の試用期間後、自由に会員に登録いただけます。お買い求めはJ&Pの各店で。または、現金書留でJ&P HOT LINE事務局までお申し込み下さい。

■申込先

〒556 大阪市浪速区日本橋5-6-7 上新電機株式会社
J&P HOT LINE 事務局宛 TEL (06)632-2521

■利用料金について

入会金／3,000円（スタータキット購入の代金から充当されます。）
接続料／3分あたり20円（アクセスポイントまでの電話代は含みません。）

スタータキット 申込書

お名前		お電話番号	（　　）　―
ご住所	〒		

お申込品　①スタータキット（ソフトなし）￥3,000

●パソコン/ワープロ通信ネットワークサービス
J&P HOT LINE

▼万全のサポート体制で全国をネットするパソコンの大型専門店 J&P チェーン

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03) 496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
ワープロランド	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
阪急三番街店	大阪市北区芝田1-1-3阪急三番街B1	☎(06) 374-3311
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3-204千里サンタウン3F	☎(06) 834-4141
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
西宮店	兵庫県西宮市河原町5-11	☎(0798)71-1171
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221

T4910217901549 雑誌 02179-1